國立交通大學

多媒體工程研究所

碩士論文

以粒子為基礎的物體融化模擬與流動控制

A Particle-based Simulation for Object Melting and Flowing Control

研究生：羅建凱

指導教授：莊榮宏　教授

林文杰　教授

中華民國 九十九 年 九 月

以粒子為基礎的物體融化模擬與流動控制

A Particle-based Simulation for Object Melting and Flowing Control

研究生：羅建凱　　Student：Chien-Kai Lo

指導教授：莊榮宏　　Advisor：Jung-Hong Chuang

林文杰　　Wen-Chieh Lin

國立交通大學
多媒體工程研究所
碩士論文

A Thesis
Submitted to Institute of Multimedia Engineering
College of Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
in

Computer Science

September 2010

Hsinchu, Taiwan, Republic of China

中華民國九十九年九月

以粒子為基礎的物體融化模擬與流動控制

研究生：羅建凱　　　　　指導教授：莊榮宏 博士

林文杰 博士

國立交通大學

資訊學院多媒體工程研究所

摘要

我們利用光滑粒子流體動力學法(Smoothed-Particle Hydrodynamics)來模擬物體融化，並且使用一個具有回饋機制的控制演算法來操控融化的流體。在納維-斯托克斯方程式(Navier-Stokes equation)中加入計算彈性壓力的力量，讓我們可以模擬出同時具有固態與液態特性的流體。我們在粒子間模擬熱能傳導的行為，利用模擬後的溫度自動來決定黏滯性與彈力的模擬係數。此外，我們設計了一套直覺化的控制機制來幫助使用者設計融化的過程。使用者可以直接在場景表面上繪製融化的路徑與融化後流體的形狀。論文的成果顯示我們可以比之前的方法保有更多融化過程中的細節，且在控制機制的幫助之下，使用者可以很容易的設計想要的物體融化過程與成果。

A Particle-based Simulation for Object Melting and Flowing Control

Student: Chien-Kai Lo

Advisor: Dr. Jung-Hong Chuang
Dr. Wen-Chieh Lin

Institute of Multimedia Engineering
College of Computer Science
National Chiao Tung University

ABSTRACT

We purpose an SPH method for object melting simulation and a feedback control algorithm to manipulate the melting flow. By adding an elastic stress term to the Navier-Stokes equation, we can simulate the melting behavior which has both fluid and solid features. We develop a thermals transfer system to simulate the heat flow between particles, and based on the temperature of particles to determine the viscosity and elastic stress coefficient automatically. Also, we introduce a feedback control algorithm which can be used to design the flowing behavior intuitively. Users can draw flowing path directly on the surface to design the shape of melting fluid. Our experiments shows that the proposed approach can preserve more details of melting than previous methods, and be useful for authority a desired a melting scene.

Acknowledgments

I would like to thank my advisors, Professor Jung-Hong Chuang and Wen-Chieh Lin, for their guidance, inspirations, and encouragement. I am also grateful to my senior colleague, Tan-Chi Ho, for his advices and suggestions on my thesis research. And then I want to express my gratitude to all my colleagues in CGGM lab: Jau-An Yang and Ta-En Chen discusses with me on research issues and helps me figure out the questions, Ying-Yi Chou and Chien Yun give me several thesis advices and some necessary information, Chia-Ming Liu and Yu-Chen Wu often share some useful programming skills and knowledge with me, Tsung-Shian Huang help me understand much rendering knowledge, and all my junior colleagues' kind assistances. Also, I want to thank my dear girlfriend Yi-Yu Lin for her support through all these years. Lastly, I would like to thank my parents and my sister for their love, and support. Without their love, I could not pass all the frustrations and pain in my life.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

Physics-based simulation has been developing in Computer Graphics field for many years. In particular, fluid simulation has been playing an important role in those works. With the help of these photorealistic results, more special effects can be designed by artists, even it is not exist in the reality. Although many stunning results had been presented in recent years, there are still many fields need to be improved. Such as the fluid detail preserving, intuitive fluid control method, and real-time rendering applications. These topics still remain a great challenge for researchers. This thesis focus on the object melting behavior and flowing control. The melting process is a complex fluid behavior in real world, this natural phenomenon including the flowing of the melting fluid, phase change between solid and fluid, and the heat transfer in the simulation elements.

In the past few years, many different methods had been purposed which may deal with the melting manner. But there are several problems remaining in previous works. First, the previous simulation didn't capture the detail of fluid, such as the droplet flow along the candle surface as in Figure 1.1. Second, rare of them considered the thermal radiation effect, which is a dominant part in thermology. And third, those methods were lack of control mechanics, thus it was hard for animators to design the way for object melting and the shape they want to achieve.

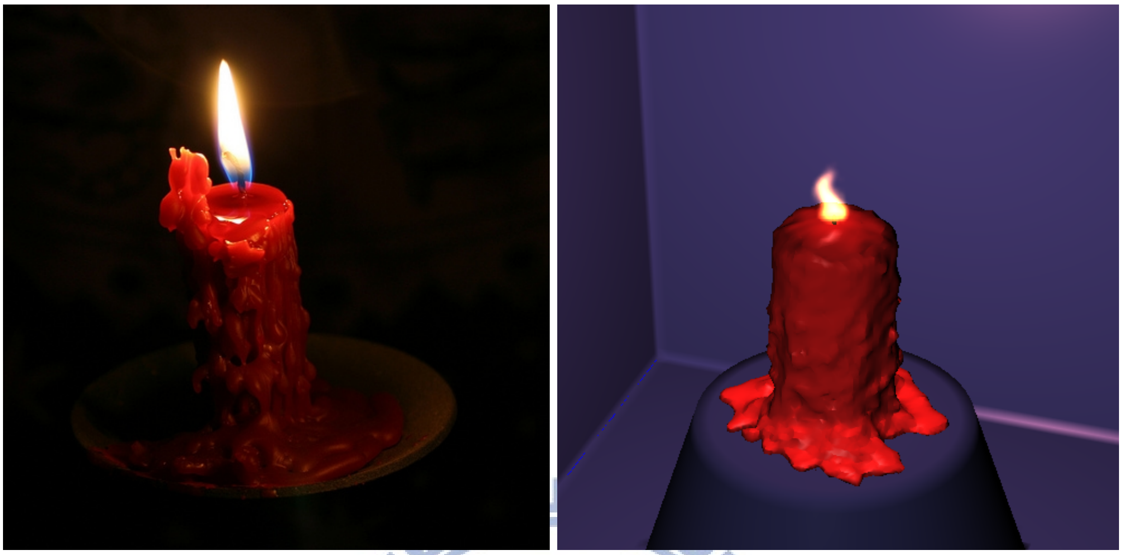


Figure 1.1: Real photo of candle melting vs. Our candle melting result

In our method, We develop a particle-based fluid solver and a physics-inspired heat transfer system to solve the object melting problems as mention above. Also, we provide a straightforward fluid control system, which allows animators to design the flowing pattern and the melting paths. For the purpose of preserving more melting detail, we hire an adaptive particle sampling scheme, enabling us to allocate the available computational resources on the regions of interest dynamically.

1.1 Contribution

The contributions of our melting simulation can be summarized as follows:

- Use an adaptive sampling scheme to enrich the melting detail without using too much particle numbers.

- A physics-inspired thermal transfer system for naturally thermal exchange behavior.

- Provide an intuitive controlling framework, which allows user to design the melting flow and the final shape easily.

- May perform a wide range of material types from high viscoelastic fluid to normal Newtonian fluid.

1.2 Organization of the thesis

The rest of the chapters are organized as follows. Chapter 2 gives a literature review for SPH fluids, melting and flowing ,and related fluid control approaches. Chapter 3 illustrates the main algorithm in our method, includes the particle-based simulation framework, thermal transfer simulation in fluid elements, adaptively particle sampling scheme in object melting, and a feedback control of fluid behavior. Chapter 4 shows the experimental results of our approach. Lastly, we summarize a conclusion and discuss the future work in chapter 5.

CHAPTER 2

Related Work

In this chapter, we briefly describe the related works by several parts: SPH-based fluids, melting and flowing, fluid control system.

2.1 SPH-based Fluids

The Smoothed Particle Hydrodynamics(SPH) method, originally devised for the simulation of stars [Mon05], was first used by Desbrun et al. [DC96] in computer graphics field to solve the interaction forces of large deformation. This method can handle complex large topological deformation and satisfies the mass conservation equation easily.

Later, Müller et al. [MCG03] introduced a particle-based fluid simulation scheme for interactive applications. By deriving the equation of pressure, viscosity and surface tension term, and using three different smoothing kernels (Figure 2.1), they show that SPH can simulate the fluid behavior appropriately. Adams et al. [APKG07] provides an adaptively sampled particle scheme, that performs the particle merge and split operation based on the extended local feature size(ELFS). The classical local feature size of a point **x**,**LFS(x)**, on a (closed) manifold **M** is

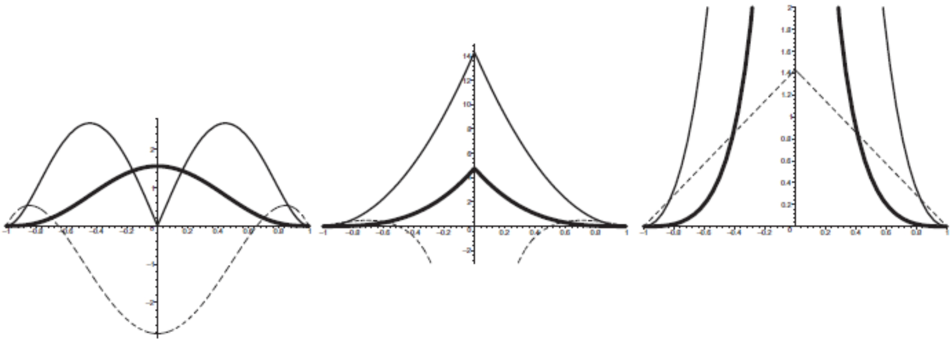


Figure 2.1: Three smoothing kernels W_{poly6}, W_{spiky} and $W_{viscosity}$ (from left to right). The thick lines are the kernels, the thin lines are the gradients in the direction towards the center, and the dash lines are the Laplacian [MCG03]

.

defined as the distance of **x** to the closest point on the manifold's medial axis. As show in Figure 2.2, they use particle approximation of the medial axis to compute the ELFS value. Red regions in Figure 2.2 (c) have a lower feature size and should be sampled densely while blue regions have a larger feature size and can be sampled coarsely. The extended local feature size of **x**, **ELFS(x)**, is defined as the minimal sum of the distance from **x** to the point **y** on the manifold and the classical local feature size at point **y** (Equation 2.1). By utilizing the ELFS value, the region with different local feature level can be clearly classified. They used larger particles in the region has larger ELFS value, which means the region is inside the volume or near a thick flat surface, to reduce the number of particles. Thus, the simulation speed may get a significant improvement.

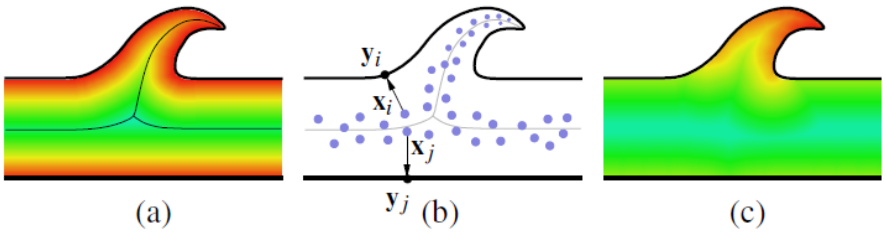


Figure 2.2: Extended local feature size [APKG07].

$$ELFS(\mathbf{x}) = \min_{\mathbf{y} \in \mathbf{M}} \{\|\mathbf{x} - \mathbf{y}\| + LFS(\mathbf{y})\}, \mathbf{x} \in \nu. \tag{2.1}$$

Figure 2.2 (a) shows the fluid volume with the medial axis. The color shows the distance to the fluid surface at any given position. Figure 2.2 (b) approximates the medial axis with particles. And Figure 2.2 (c) shows the color-coded ELFS value.

Moreover, there are several studies on more complicated fluid phenomena. Lenaerts et al. [LAD08] introduced an SPH algorithm to simulate the flow through the porous deformable material and showed some interesting demos, such as the simulation of sponge-like elastic bodies and water-absorbing sticky cloth. Lenaerts et al. [LD09] also simulated the interactions between fluids and granular materials by purposing a unified SPH framework. They treat the granular material as a continuous volume sampled by particles, which can be animated like SPH fluids. Solenthaler et al. [SP09] enforced incompressibility by purposing a prediction-correction scheme to solve particle pressure. They avoided the computational cost of solving a pressure Poisson equation, and achieved an order of magnitude speed-up compared to the commonly used weakly compressible SPH.

SPH has been more and more widely used in fluid simulation. By representing the fluid with a set of particles, we may access individual material properties directly and move the particles according to the Navier-Stoke equation easily. A detailed and extensive introduction to SPH can be found in [Mon05].

2.2 Melting and Flowing

Object melting and flowing has many interesting effects, and it is still remaining a challenging field for researchers. Carlson et al. [CMVHT02] modified the Marker-and-Cell (MAC) algorithm to compute the fluid variable and arbitrarily high viscosity. They also introduced a heat diffusion equation to handle the change in temperature, and varies the material's viscosity according to the temperature. Keiser et al. [KAG+05] proposed a unified model to solve the deformation of solids and fluids, by combining the solid mechanics with Navier-Stokes equation in a Lagrangian form. As shown in Equation 2.2, they applied the same concept as heat transfer equation in [CMVHT02] but in a SPH manner.

$$\frac{DT_i}{Dt} = \eta \sum_j (T_j - T_i) \nabla^2 W(\mathbf{r} - \mathbf{r}_j, h) \frac{m_j}{\rho_j} \tag{2.2}$$

Paiva et al. [PPLT06] [PPLT09] simulated the object melting effect by adding a viscoplastic stress tensor term to the traditional Lagrangian fluid formulation. They used the same SPH concept in [MCG03] but with different smoothing kernels. Instead of three different smoothing kernels, one unified smoothing kernekl and a XSPH correction scheme is used to reduce the particles instability due to the arbitrarily small distance between particles. They modeled the stress tensor with the Generalized Newtonian Fluid model, and simplified the viscosity control by introducing a new rheological parameter called jump number. Chang et al. [CBL+09] animated the viscoelastic fluid by adding an elastic stress to the momentum conservation equation (a modified version of Navier-Stokes equation).Their method also showed the ability to simulate the object melting and flowing behavior by using the concept similar in [CMVHT02] where the viscosity coefficient μ and elastic stress coefficient μ_s are influenced by the temperature. All the methods mentioned above solve the heat diffusion equation and varied the viscosity depending on the temperature, but none of them considered the heat radiation effect, which plays an important role in heat transfer system. Wei et al. [WLK03] employed a linear 3D cellular automata approach on the solid volumetric models to animate the melting process. By reducing the physical simulation rule, they may animate the smooth fluid behavior in an interactive rate.

Maréchal et al. [MGG$^+$10] developed a physically based heat transfer simulation system, to synthesize realistic winter sceneries. They defined the environmental model and three major types of thermal transfers in different environmental elements. Figure 2.3 shows the thermal transfer they had taken into account, including heat conduction, heat convection, and heat radiation in different materials. Our thermal transfer system is similar to their's, but without the heat convection term that is less important in normal object melting.

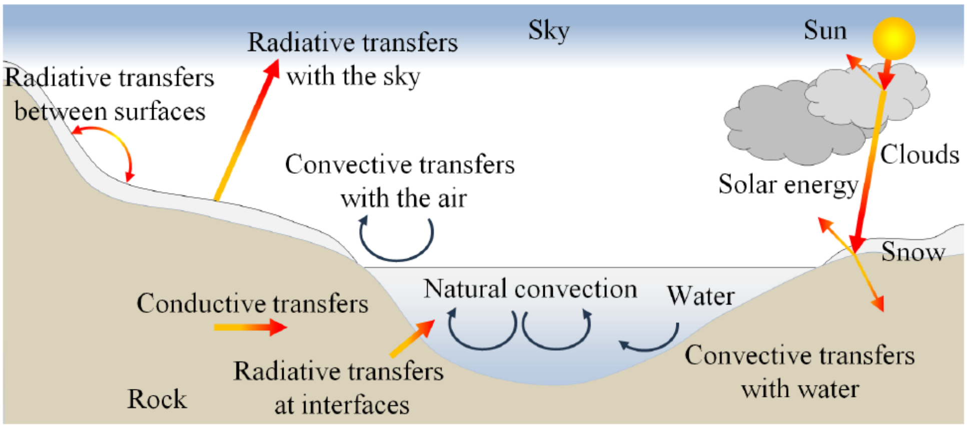


Figure 2.3: Thermal transfer in [MGG$^+$10]

2.3 Fluid Control System

Simulation of realistic fluid behavior has been always a hot field in computer graphics. Recently, more and more researchers have focused on the fluid control mechanism for artistic reasons in artist. Foster et al. [FF01] was the first to apply control scheme on fluids. They produced 3D parametric space curves to locally control the velocity and pressure of the fluid flow. Rasmussen et al. [REN$^+$04] defined four types of control particles according to the liquid variables, including velocity, viscosity, level set, and the velocity divergence particles, where velocity particles controlled the movement of the liquid interface; viscosity particles for viscous melting behavior; level set particles is used to modified the level set representation in liquid interface; divergence particles modeled the liquid expansion and contraction effects. Treuille et al. [TMPS03] enforced the smoke to match the given velocity and density keyframes by building an optimization technique to solve the control parameters. McNamara et al. [MTPS04] used the adjoint method to improve the efficiency of solving the nonlinear optimization problem. The key to their approach is to utilize the linear duality for variables substitution. The adjoint method exploited this powerful aspect of duality to accelerate the gradient calculation in optimization process. Fattal et al. [FL04] controlled the smoke shape with a sequence of predefined target smoke states. They added two terms to the standard flow equations : driving force term and smoke gathering term. While the driving force compelled the fluid to a particular smoke shape, the gathering term gathered the fluid to maintain the density value. Instead of solving the control forces optimization problem, Hong et al. [HK04] created a geometric potential field to exert the external force, and coerce the fluid to form the target shape. Shi et al. [SY05] controlled the liquids to match with the rapidly changing targets by building two different external force fields. The first one is a feedback force field applied to both shape and velocity to compensate the discrepancies with the target field. The second one is the gradient field of an adaptive geometric potential field defined by the target object shape and skeleton information. Compared to the original geometric potential field in [HK04], their adaptive potential field could handle different complex target shape with appropriate force magnitude. Later, Thürey et al. [TKPR06] introduced a control particles system to control the liquid without losing the detail part. As show in Figure 2.4, they first generated control particles set using some predefined criteria, then computed the attraction force and velocity force acting on the general fluid elements separately. In the end, they combined the control forces with weighting sum to evaluate the final forces for each element. In order to alleviate the virtual viscosity effect, they also applied high and low pass filter to preserve the detail motion. More recently, Dobashi et al. [DKNY08] developed a cloud formation control system. They provided a geometric potential field similar to the previous method [HK04] to solve the horizontal shape, but used a different feedback control scheme to control the cloud vertical height. Instead of feedback to the force strength, their feedback system controlled a water vapor supplier and a latent heat controller,

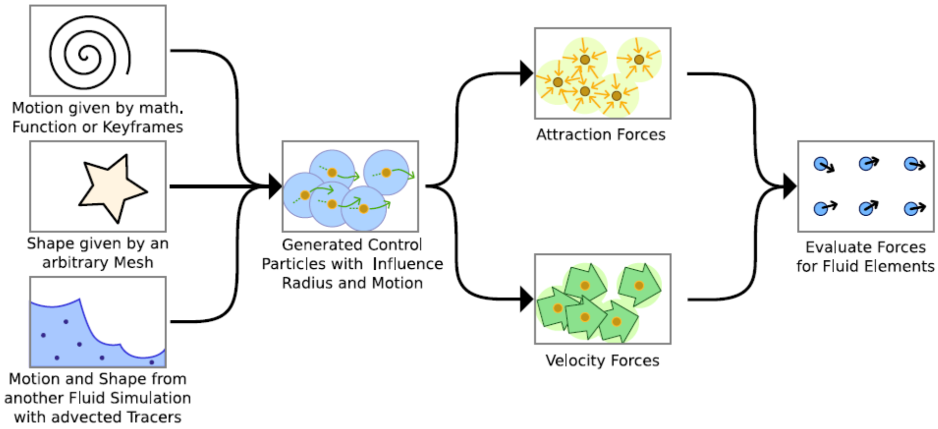


Figure 2.4: Particle-based control system [TKPR06]

allowing them to deal with the phase change effects and the cloud generation.

Though there are many fluid control methods, none of them had dealt with object melting and flowing problem, which is more complex because of the heat transfer involved in fluid elements and the melting behavior.

CHAPTER 3

Simulation and Control Framework

3.1 Overview

The aim of our algorithm is to simulate the object melting naturally, and also provide a control system to design the melting scene. First, the melting behavior should be triggered when the temperature reaches the melting point, and the material could transform from solid into fluid smoothly. Second, the simulation ought to preserve the detail drops flow on the surface. Third, although the simulation follows the physics law, animators still need certain control ability to design the flow as they wish.

Figure 3.1 is the system flow of our algorithm. In the initial stage, user can author a melting scene as a input to our control system. In the simulation part, we first solve the Navier-Stokes equation using SPH method. Also, by adding an elastic force term to the momentum conservation equation, we are able to simulate the high viscous fluids and viscoelastic fluids. After solving SPH, we simulate the thermal transfer between particle elements. We have modeled

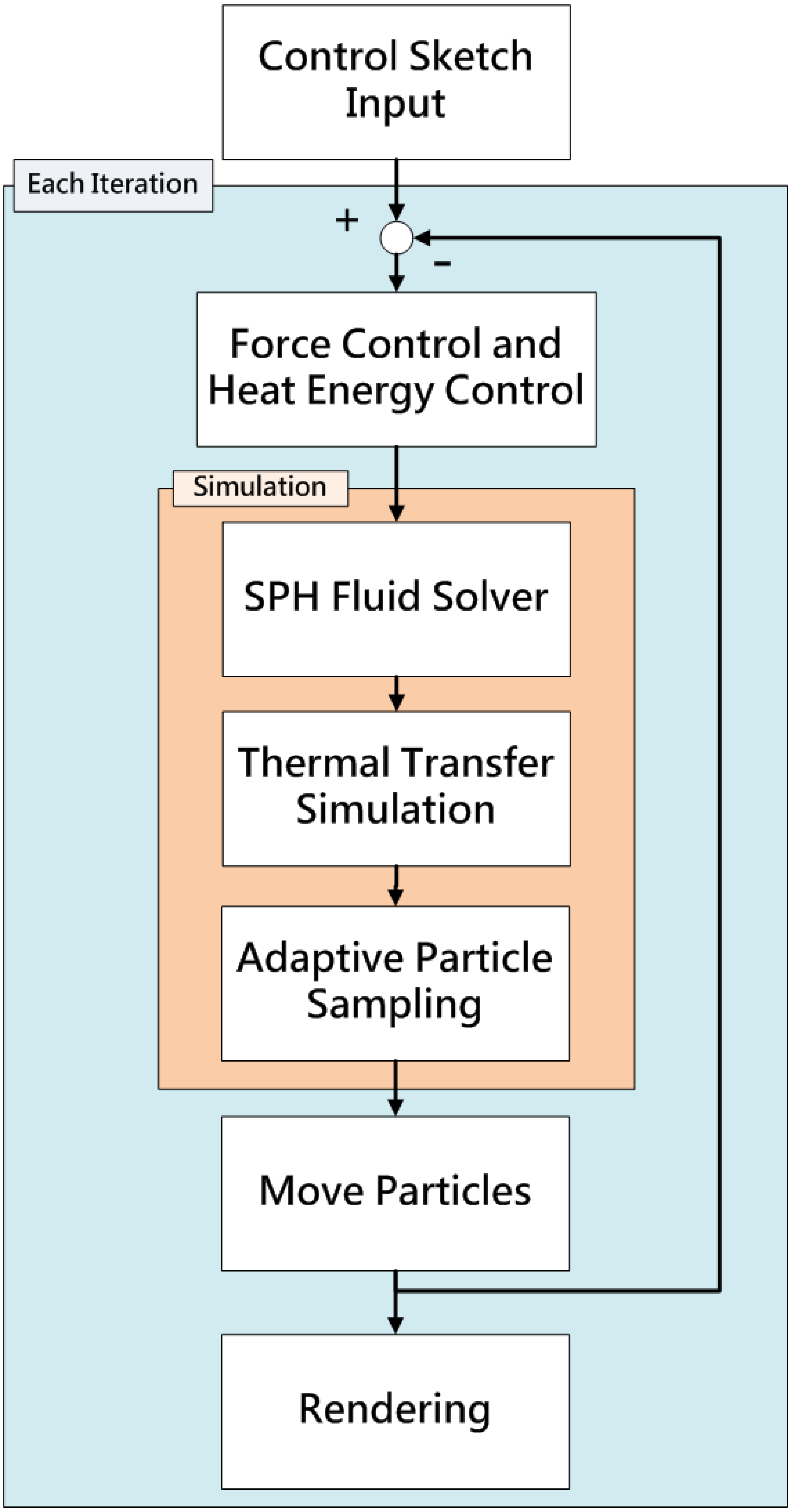

Control Sketch Input
Each Iteration
+
-
Force Control and Heat Energy Control
Simulation
SPH Fluid Solver
Thermal Transfer Simulation
Adaptive Particle Sampling
Move Particles
Rendering

Figure 3.1: System flow of our algorithm

three types of thermal transfer methods: thermal radiation, thermal conduction, and thermal dissipation. According to the temperature of the particle, the phase change effect can be triggered by changing the viscosity and elastic coefficients. Besides, we employ an adaptive particle sampling scheme to improve the efficiency, allowing us to use more computational resource for the melting region. Based on the output of the simulation part, we feedback to the control mechanism to adjust the force field automatically. Then, we can move the particles by an Euler scheme. In the end, we reconstruct the surface by the marching cube algorithm and render the polygon mesh for output images.

3.2 Fluid Simulation Framework

In this section, we describe the physics framework in the fluid solver. Figure 3.2 shows the system flow of fluid solver. We first solve the Navier-Stokes equation with viscoelastic force in the SPH manner and an adaptive sampling scheme.

3.2.1 SPH formulations

SPH is an interpolation method defined on the particle system. In SPH, fluids are represented by discrete particles and all the fluid properties are carried on those particles. As shown in Equation 3.1, one can evaluate a particular quantity A at any position r according to the weighted sum of contributions from the neighboring particles j.

$$A(r) = \sum_j A_j \frac{m_j}{\rho_j} W(\mathbf{r} - \mathbf{r_j}, h) \tag{3.1}$$

Where r and r_j represent a any given position and the neighboring particle j position, m_j means the mass define on the neighboring particle j, ρ_j is the density define on the particle j, A_j is a scalar value define on the particle j. The W is the smoothing kernel with smoothing length h. The gradient and the Laplacian of scalar A can be calculated by using the following gradient and the Laplacian of the smoothing kernel W (Equation 3.2, Equation 3.3):

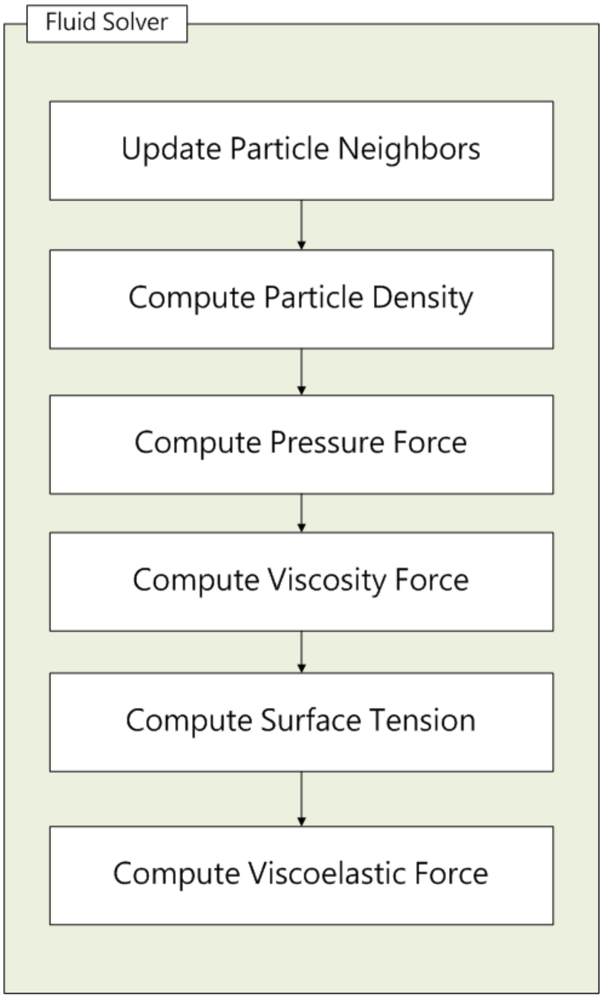


Figure 3.2: Fluid solver flowchart

$$\nabla A(r) = \sum_j A_j \frac{m_j}{\rho_j} \nabla W(\mathbf{r} - \mathbf{r_j}, h) \tag{3.2}$$

$$\nabla^2 A(r) = \sum_j A_j \frac{m_j}{\rho_j} \nabla^2 W(\mathbf{r} - \mathbf{r_j}, h) \tag{3.3}$$

Before calculating the force fields, we need to have the density and the pressure information for every particle. The density ρ (in Equation 3.4) and the pressure (in Equation 3.5) of the particle i are calculated as in [MCG03]:

$$\rho_i = \sum_j m_j W(\mathbf{r}_{ij}, h_i) \tag{3.4}$$

$$P_i = k(\rho_i - \rho_0) \tag{3.5}$$

where $r_{ij} = r_i - r_j$, k corresponds to the gas constant, ρ_i is the density of particle i and ρ_0 is the rest density. The pressure of each particle is given according to the ideal gas state equation [DG96].

The fluids in SPH method need to satisfy the Navier-Stokes equation, including the mass conservation equation (Equation 3.6) and the momentum conservation equation. (Equation 3.7). We add an elastic stress tensor force to the traditional momentum conservation equation in order to model the high viscosity and viscoelastic fluids properties.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \tag{3.6}$$

$$\rho(\frac{D\mathbf{v}}{Dt}) = -\nabla p + \mu \nabla^2 \mathbf{v} + \mathbf{f}_{surface} + \mathbf{f} + \mathbf{f}_{elastic} \tag{3.7}$$

There are five force field in the right hand side of Equation 3.7. $-\nabla p$ is the pressure term, $\mu \nabla^2 \mathbf{v}$ is the viscosity, $\mathbf{f}_{surface}$ is the surface tension force field, f is often thought as the external force (ex: gravity force, $f = \rho g$), and $\mathbf{f}_{elastic}$ is the elastic strain tensor force field which will be detailed in subsection 3.2.2. Following the adaptively sampled particles scheme in [APKG07], we use a symmetric fluid force to avoid asymmetrical force distribution. We will explain our adaptive sampling system in the next section. The fluid force acting form particle p_j on p_i in Equation 3.7 can be obtained as follows:

$$\mathbf{f}_{ij}^{pressure} = -\nabla p = -\frac{m_i}{\rho_i}\frac{m_j}{\rho_j}\frac{(P_i + P_j)}{2}(\nabla W(\mathbf{r}_{ij}, h_i) + \nabla W(\mathbf{r}_{ij}, h_j)) \tag{3.8}$$

$$\mathbf{f}_{ij}^{viscosity} = \mu \nabla^2 \mathbf{v} = -\frac{m_i}{\rho_i}\frac{m_j}{\rho_j}\frac{(\mathbf{v}_j - \mathbf{v}_i)}{2}(\mu_i \nabla^2 W(\mathbf{r}_{ij}, h_i) + \mu_j \nabla^2 W(\mathbf{r}_{ij}, h_j)) \tag{3.9}$$

$$\mathbf{f}_{ij}^{surface} = \begin{cases} -\frac{\sigma}{2}(\frac{\mathbf{n}_i}{\|\mathbf{n}_i\|} + \frac{\mathbf{n}_j}{\|\mathbf{n}_j\|})\nabla^2 W(\mathbf{r}_{ij}, h_i), & \text{if } \|\mathbf{n}_{ij}\| \geq \delta \\ 0, & \text{if } \|\mathbf{n}_{ij}\| < \delta \end{cases} \tag{3.10}$$

where $\mathbf{n}_{ij} = \frac{m_i}{\rho_i}\frac{m_j}{\rho_j}\nabla W(\mathbf{r}_{ij}, h_i)$, and P is the pressure, $r_{ij} = r_i - r_j$, $\mathbf{v}$ is the velocity vector, μ is the viscosity coefficient, σ is the surface tension coefficient, and $\mathbf{n_i}$ is the gradient of the color field as introduced in [MCG03]. If the magnitude of $\mathbf{n}$ is larger than a threshold δ, particle i is closed to the fluid surface and we need to compute the surface tension force. Moreover, we also store these particles as surface particles in order to conveniently compute the thermal radiation gathered from the outside heat sources and the thermal dissipation to the contact medium from surface.

3.2.2 Viscoelastic Fluid Model

In order to compute the elastic term in Equation 3.7, our method follows the concept in [GBO04] and [CBL$^+$09], which is to decompose the total strain into elastic and plastic component as in Equation 3.11),

$$\epsilon^{Total} = \epsilon^{Plastic} + \epsilon^{Elastic} \tag{3.11}$$

and compute each component by integrating their time derivatives. The total strain at a given time t can be formulated as in Equation 3.12.

$$\epsilon_{t+\Delta t}^{Total} = \epsilon_t^{Total} + \frac{d\epsilon^{Total}}{dt}\Delta t \tag{3.12}$$

where Δt is the time interval for each iteration. We can get the $\epsilon^{Plastic}$ and $\epsilon^{Elastic}$ by a similar form. Assuming the total strain and the plastic strain are both zero in the initial condition, we can evaluate the elastic strain rate by computing the difference of the total strain rate and plastic strain rate as in Equation 3.13:

$$\frac{d\epsilon^{Elastic}}{dt} = \frac{d\epsilon^{Total}}{dt} - \frac{d\epsilon^{Plastic}}{dt} \tag{3.13}$$

The total strain rate is given as follows:

$$\frac{d\epsilon^{Total}}{dt}\Delta t = \frac{\nabla\mathbf{v} + (\nabla\mathbf{v})^T}{2} \tag{3.14}$$

where $\nabla\mathbf{v}$ is the velocity tensor field defined on each particle and can be computed by Equation 3.15:

$$\nabla\mathbf{v}_i = \sum_j \frac{m_j}{\rho_j}(\mathbf{v}_j - \mathbf{v}_i) \otimes \nabla W(r_{ij}, h_i) \tag{3.15}$$

We can calculate the plastic strain according to the Von Mises's criterion: there is no plastic flow as long as the Frobenius norm of the elastic strain deviation ϵ' is still below to a yield threshold γ. Once the magnitude is greater than the yield point, the rate of the plastic flow is proportional to the difference of the Frobenius norm and γ.

$$\epsilon' = \epsilon^{Elastic} - \frac{Trace(\epsilon^{Elastic})}{3} I \tag{3.16}$$

$$\frac{d\epsilon^{Plastic}}{dt} = \xi \frac{\epsilon'}{\|\epsilon'\|} max(0, \|\epsilon'\| - \gamma) \tag{3.17}$$

where ϵ' is the elastic strain deviation, $Trace(A) = a_{11} + a_{22} + ... + a_{nn} = \sum_{i=1}^{n} a_{ii}$, I is the identity matrix, γ is the yield threshold, and ξ is the material's elastic decay rate. Finally, we can have the elastic strain rate from the Equation 3.13, and we can iterate the elastic strain tensor for a given time step by integrating the time derivatives. Since we get the elastic strain for each particle, the elastic stress force acting from particle p_j on p_i can be computed as follows:

$$f_{ij}^{Elastic} = \frac{m_i}{\rho_i}\frac{m_j}{\rho_j}\frac{\kappa_i + \kappa_j}{2}\epsilon_i^{Elastic} \cdot (\nabla W(r_{ij}, h_i) + \nabla W(r_{ij}, h_j)) \tag{3.18}$$

The κ_i represents the elastic stress coefficient of particle i.

3.3 Heat Transfer System

The melting process is triggered when the temperature reaches the melting point. Thus, we require a thermal transfer simulation for the heat energy flow in the fluid elements, which are SPH particles in our approach. We have modeled several types of thermal exchange types as listed here:

- Thermal radiation from heat source to surface particles,
- Thermal conduction from particle to its neighbor particles,
- Thermal dissipation from surface particles to cold air.

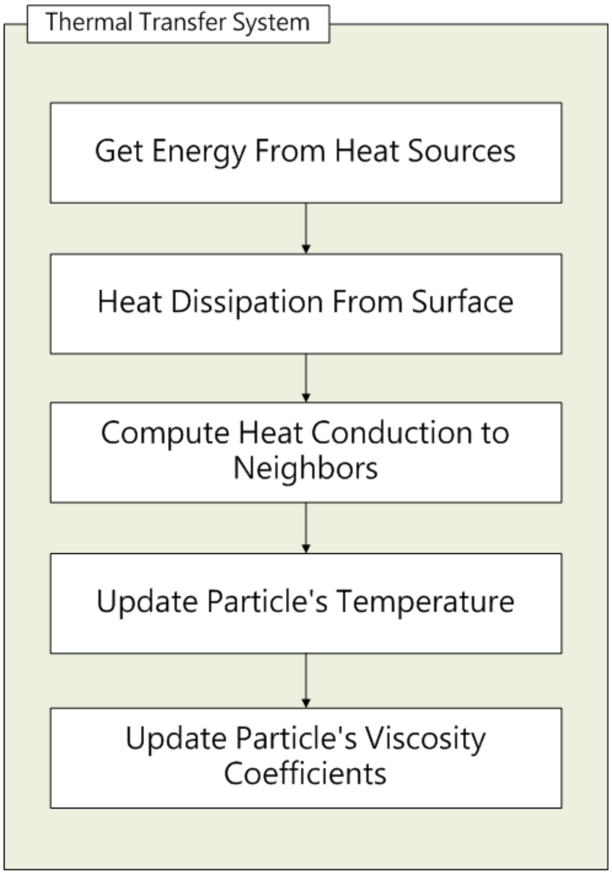

Thermal Transfer System
Get Energy From Heat Sources
Heat Dissipation From Surface
Compute Heat Conduction to Neighbors
Update Particle's Temperature
Update Particle's Viscosity Coefficients

Figure 3.3: Thermal transfer system flowchart.

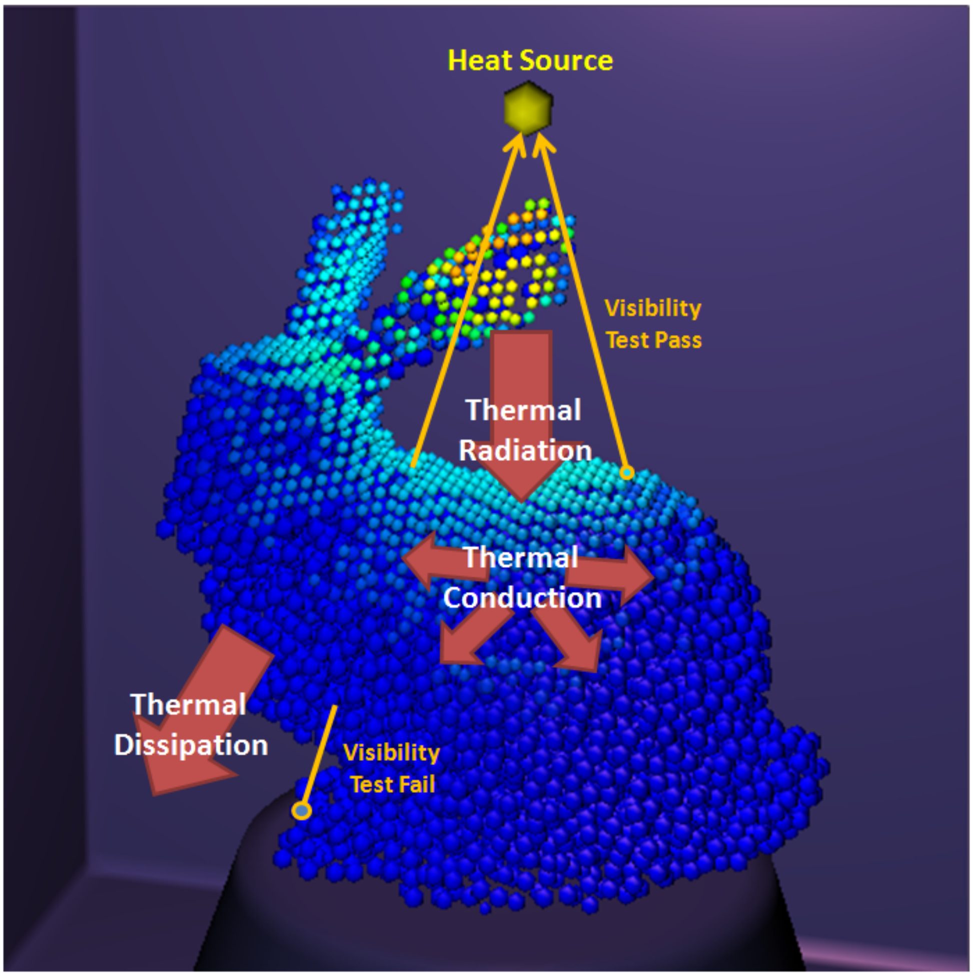


Figure 3.4: Thermal simulation in our approach.

The thermal information is stored on each particle and heat energy can be exchanged with the neighborhood in the particle supporting range. Every particle has the heat properties as follows: total heat energy, temperature, absorb energy, latent heat threshold, conductive rate, emissive rate, and adopt heat energy coefficient. The total heat energy shows the current heat energy amount in this particle and can be converted into temperature using Equation 3.19:

$$T_i = \frac{Q_i}{m_i C_i} \tag{3.19}$$

where T_i is the temperature, Q_i is the heat energy stored in particle i, and C_i is the specific heat capacity. The absorb energy and latent heat threshold are used to interpolate the viscosity and the elastic stress coefficients for the melting process from solid to liquid. When the temperature reaches the melting temperature, which invokes the phase change condition, particles start absorbing energy to destruct the solid structure to become liquid. The viscosity and elastic stress coefficient are inversely proportional to the ratio of absorb energy and latent heat. Conductive rate defines the conduction between particles and emissive rate represents the emissivity from the surface particles to the air. Finally, adopt heat energy coefficient is used to adjust the amount of heat energy obtained from heat sources per iteration.

Besides, to simulate the thermal radiation and the thermal dissipation from the object surface, we have to identify the surface particles first. We make use of the gradient of the color field in Equation **??**, which also represents the surface normal direction. By detecting whether the magnitude exceeds a given threshold, we may extract the surface particles and the surface normal as shown in Figure 3.5.

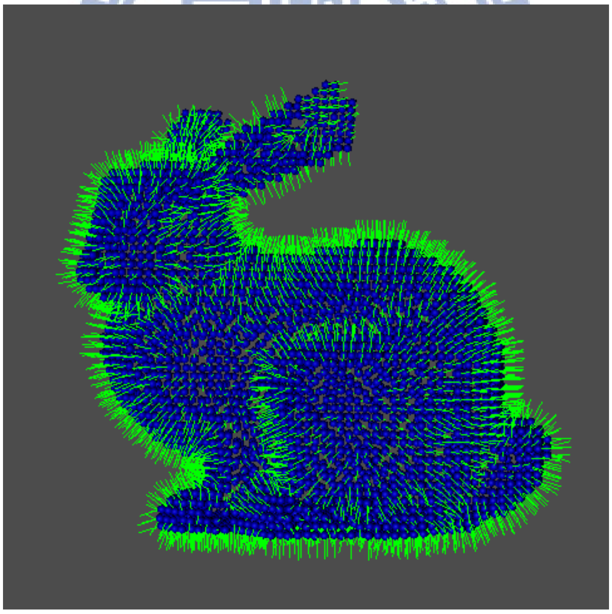


Figure 3.5: Surface particle and its normal.

Thermal Radiation transmits energy from heat sources to the surface particles in a radiative way. In this simulation framework, we only consider the direct radiative ray from the sources.

Indirect radiation (multi-bounce of radiative ray) is ignored in our approach since its effect is insignificant. Therefore, we first detect the visibility by ray casting between surface particles and the heat sources. If the particle pass the visibility test, we compute the angle between the surface normal vector and the vector from particle to heat source position (see Figure 3.6). We than calculate the energy value depending the Equation 3.20.

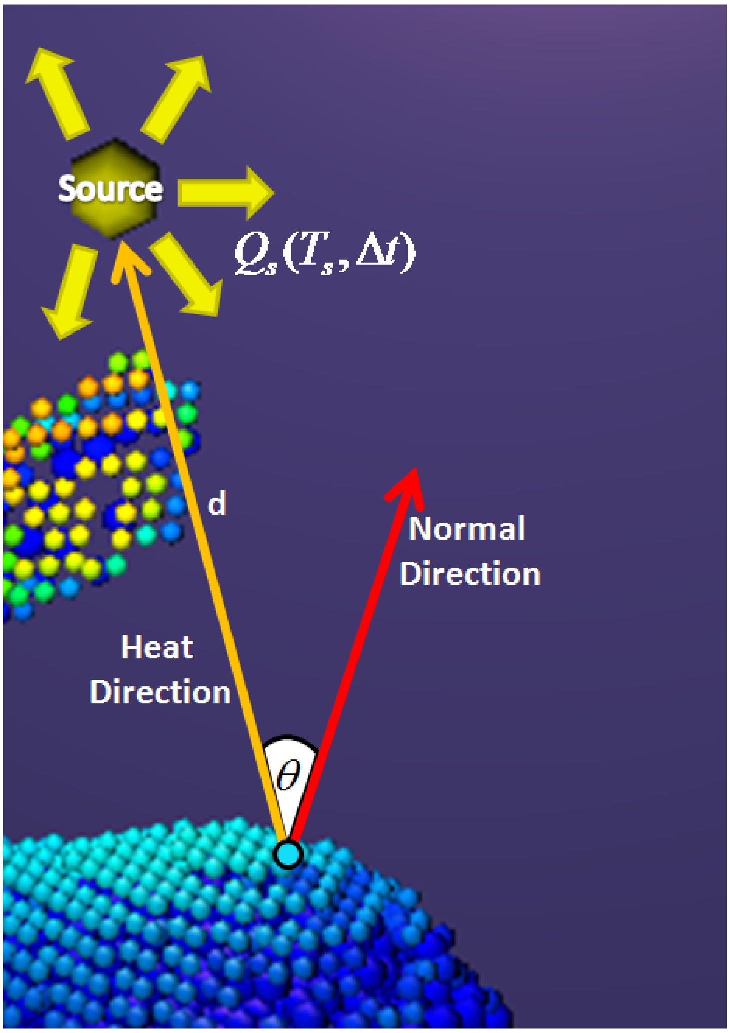


Figure 3.6: Thermal radiation diagram

$$\Delta Q = \sum_s \delta \frac{m}{d^2} \cos\theta Q_s(T_s, \Delta t) \tag{3.20}$$

where ΔQ is the energy variation in each iteration, δ is the adopt heat energy coefficient, d is the distance between the source and the particle position, $\cos\theta$ is the angle we mentioned above, Q_s is a function representing the heat energy value generated by the heat source. $Q_s(T_s, \Delta t) \propto T_s\Delta t$ in our implementation.

We have modeled two kinds of heat sources in our simulation. The first one is the point heat source as in the Figure 3.4. Another one is the area heat source, we modeled this as a plane which provides the heat energy for one side. The area heat source is used to model the heat source like a hot pan or hot ground.

Thermal Conduction diffuses the energy between particle and particle. According the physics of thermology, the heat energy flows from region of high temperature to region of lower temperature until it reaches the equilibrium state. Therefore, we model the heat conduction only if the particle temperature is higher than its neighborhood and the diffusion range is defined as the particle's supporting range. The conduction is proportional not only to the ΔT, but also to the mass and the inverse of distance between particles, as shown in the following Equation 3.21 and 3.22:

$$Q_c = \lambda_i Q_i \frac{\frac{m_j(T_i - T_j)}{d_{ij}}}{W_{sum}} \tag{3.21}$$

$$W_{sum} = \sum_j \frac{m_j(T_i - T_j)}{d_{ij}} \tag{3.22}$$

where Q_c is the heat energy through conduction, d_{ij} is the distance between particle i and particle j, and λ is the conductivity we have mentioned above. We distribute the energy flow fairly to the neighborhood particles. Equation 3.21 computes the heat energy conducts from particle i to its neighborhood particle j, and Equation 3.22 evaluates the weighting of conduction by considering the distance, particle mass, and the difference of temperature of two particles.

Thermal Dissipation is another thermal radiation in which the energy is radiated from the surface to the outside medium. The general equation of thermal radiation can be expressed as

Equation 3.23 because most of the materials in nature are considered as grey bodies, we need to consider the emissivity ϵ. This factor has to be multiplied with the radiation spectrum formula before integration. For simplicity, we can simplify the factor as a constant similar to [MGG$^+$10] did.

$$Q = \epsilon \cdot \sigma \cdot A \cdot T^4 \tag{3.23}$$

Equation 3.23 is the equation of the general thermal radiation in physics, where factor σ is the Stefan-Boltzmann constant, ϵ is the emissivity, A is the radiating surface area, and T is the object temperature. Based on Equation 3.23, we formulate the thermal dissipation in our simulation as the following form:

$$Q_{emissive} = \epsilon\sigma V_p(T_p^4 - T_{air}^4) \tag{3.24}$$

Equation 3.24 computes the heat energy dissipate from the surface to the contact medium. We use the particle volume V_p to approximate the area term since it is much easier to compute in our approach.

After all the computation of energy transfer, we may update the temperature of particles through Equation 3.19, also update the viscosity according to the ratio of energy absorbtion and latent heat value.

3.3.1 Previous Thermal Diffusion Method

In the previous methods for object melting problem ([CMVHT02], [PPLT06], [CBL$^+$09], [WLK03]), neither of them had implemented the thermal radiation effect, and some of them use the SPH interpolation scheme to simulate the thermal diffusion between particles. But in the real world, thermal radiation is actually a domination part in object melting. And the thermal conduction usually propagate the heat energy much slower than the radiation. In their demos, we found it is hard to capture the detail drops flowing and the melting behavior caused by the exposure to heat sources. Also, the thermal radiation allows us to determine the melting region and the melting speed in a natural way, but previous methods require users to define particle temperature in other rules.

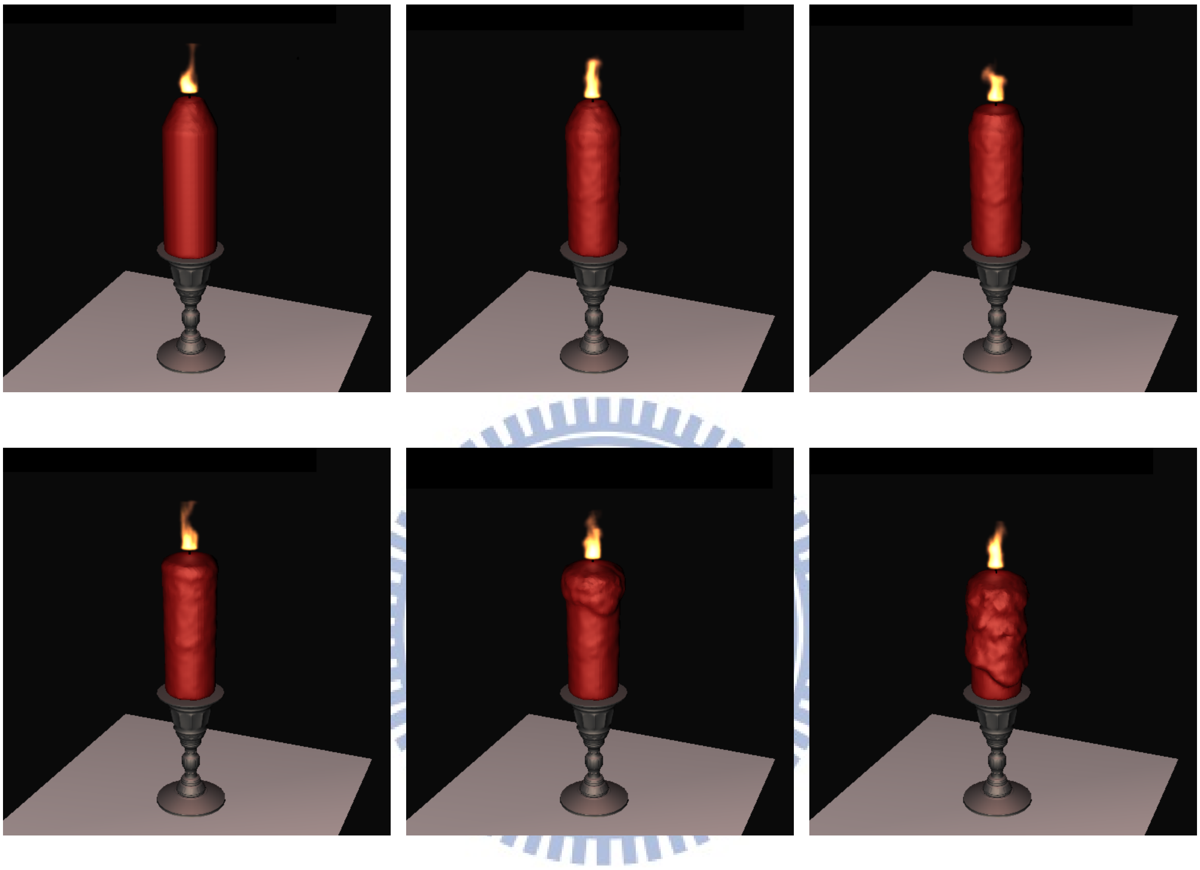


Figure 3.7: Candle melting without thermal radiation and dissipation

Figure 3.7 shows the melting simulation involving only thermal conduction, which is done by using the SPH interpolation as in [CBL^{+}09]. The temperature is increased by adding a constant value per iteration around the heat sources.

3.4 Adaptive Particle Sampling

The goal in adaptive particle sampling is to reduce the particle number in less important regions. Therefore, the computational resources could focus on the regions of interest; for example, the melting particles. Our adaptive scheme is basically based on the works in [APKG07], but we additionally take temperature influence into account. We adapt the particle sampling density during the simulation, and classify them into different levels l, where $l = 0, 1, 2, 3....$ The lowest level 0 particles have a mass m and support radius r. The particles in other levels have mass and support range computed according to their level as follows:

$$m_i = 2^{l_i} m \tag{3.25}$$

$$r_i = \sqrt[3]{2^{l_i}} r \tag{3.26}$$

The criterion of splitting or merging not only consider the ELFS value but also the temperature and the distance to the heat sources. After updating the ELFS and temperature value, particles are added or removed depending on the conditions below:

A particle p_i is split or merged if the following condition are all true:

- Split
  1. $L_i > 1$
  2. $ELFS(x_i) < \alpha r_i$
  3. $T_i < \frac{\alpha_T}{L_i}$ or $\|p_i - H_s\| < \frac{\alpha_d}{L_i}$
- Merge
  1. $ELFS(x_i) > \beta r_i$ or $(T_i < \frac{\beta_T}{L_i}$ and $L_i < 3)$
  2. $\|p_i - H_s\| > \frac{\beta_d}{L_i}$

where L_i is the level of particle i, T_i is the temperature of particle i, $\|p_i - H_s\|$ represents the distance between the particle and the heat source. The rest of six α, α_T, α_d, β, β_T, β_d are different coefficients respectively. The split operation happened when the ELFS value is below to a certain threshold, and the temperature is increased over the value defined by the inverse level. To avoid the split and merge operations take place too frequently due to the dramatically change in temperature, we also consider the distance between particle and heat source since the region close to heat source may has higher possibility to vary the temperature. On the other hand, the particles merge if the ELFS value or the temperature threshold satisfy our condition. We limit the merge level to prevent the excessive merge, and this level upper bound is an experimental result in our simulation.

Figure 3.8 shows the particle distribution and the reconstruct surface in the adaptive and the general version. The first column is the general sampling condition, which uses 17685 particles during the simulation iterations. The second column is our adaptive sampling scheme, the particle number can be lower at most to minimum 5994 particles, achieving a great advantage of computation efficiency. Figure 3.8 (A) are the reconstructed polygon meshes using marching cube algorithm, although the particle numbers differ greatly, the surface is still similar to each other. Figure 3.8 (B) is the particle view for our simulation, and the color from red to blue represent the ELFS value from low to high. Figure 3.8 (C) is the color-coded particle temperature, we aggressively merge the particles where the temperature is still low.

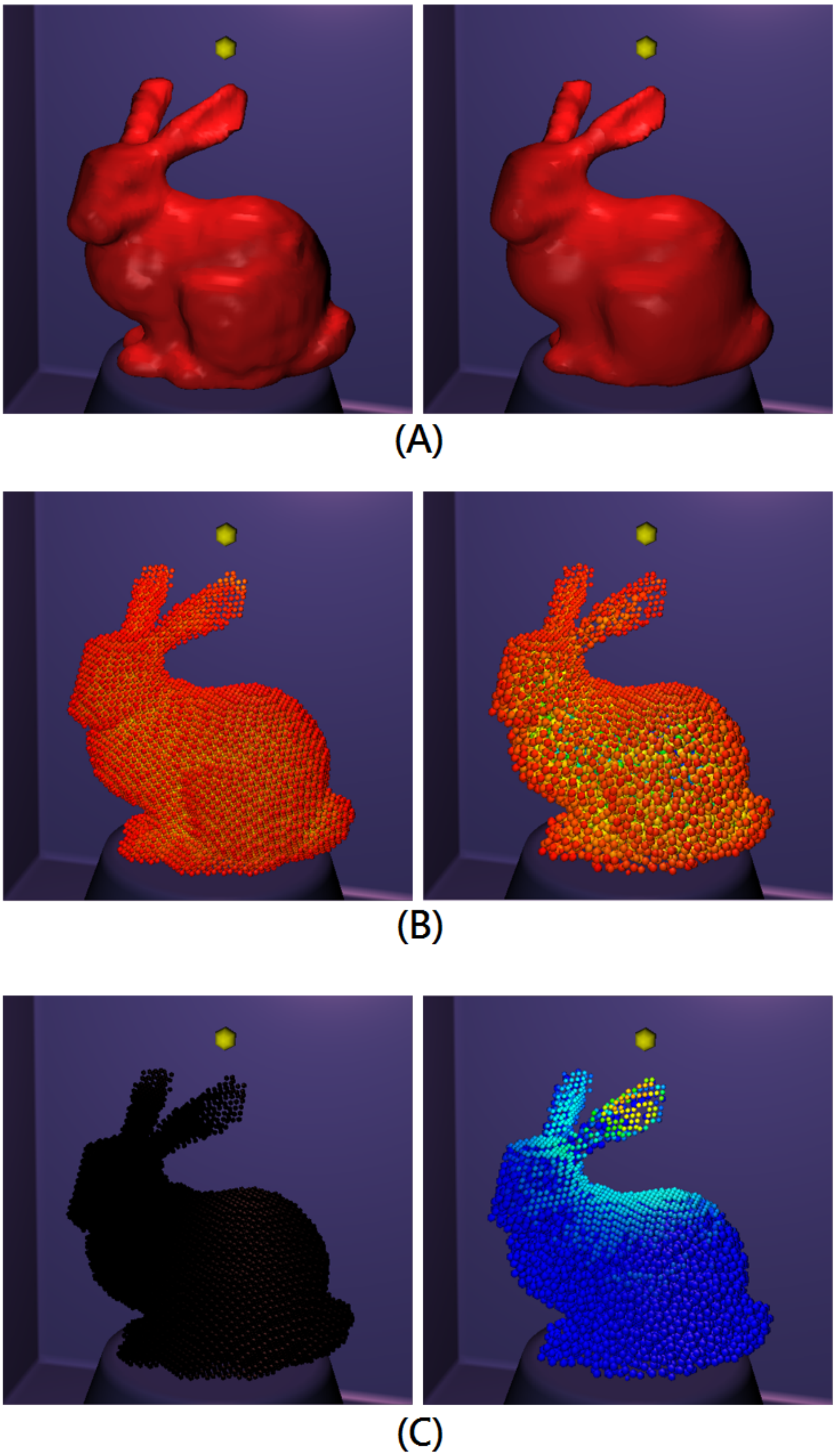


Figure 3.8: Adaptive particle sampling (on the right) compared to uniform sampling (on the left).

Our implementation of split and merge operation is a modified version of [APKG07]. Figure 3.9 (a) and (b) indicate the resampling operations. Figure 3.9 (a) shows how the split operation works. While a particle need to perform the split operation, it will split two particles place on the symmetric position of its original position. After the splitting, the original particle will be deleted. Figure 3.9 (b) indicates the merge operation. While two particles are all tagged to be merged, the new particle will be generated at the position of the centroid of two particles and the two merged particles will be deleted.

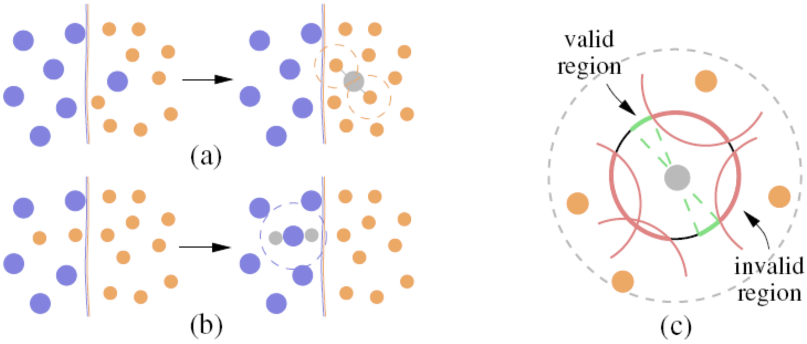


Figure 3.9: Particle resampling scheme in [APKG07].

Split When a particle p is split, the new generated particles p_1, p_2 are positioned symmetrically around p. The position of new particles should be chosen carefully to avoid the large pressure forces caused by the neighboring particles. Figure 3.9 (c) shows the valid region to sample the split particles. The possible valid region is defined as a sphere with radius d to the p's center position. We check all the neighboring particles ,which the distance between p and the neighboring particles are less than $2d$, in order to narrow down the valid regions which can place the new particles. The regions are eliminated around particle p if they are too close to the neighboring particles, which means the distance is lower than d(the red arcs in Figure 3.9 (c)). For the rest of regions become valid region if the two split particles can be placed symmetrically around the particle p position(the green arcs in Figure 3.9 (c)). If there is no valid region can be found, the distance d will be lowered down iteratively until we have a valid region. After finding the region we can sample on, we may have many possible symmetric pair candidates, and the algorithm just randomly pick one of them.

As for the parameters of particles p_1 and p_2, their levels l_1 and l_2 are decided as $l_i - 1$. The parameter about the particle's properties (e.g. conductivity, emissivity, temperature, elastic stress coefficient, viscosity coefficient, and velocity ...etc) is the same as the original particle p, and the energy store in p (e.g. absorb energy, heat energy) can be distributed evenly to p_1 and p_2 since their mass are equal.

Merge Two particles p_i and p_j can be merged if they are neighbor to each other and both of them pass the conditions we have mentioned above. A new particle p_k will be generated at the position $x_k = \frac{x_i+x_j}{2}$, and the level is $l_k = l_i + l_j$. The position should be chosen carefully to inside the fluid volume and also avoid placing too close to other particles, which may cause large pressure forces. If there is no merge pair pass the above criteria, then the merge operation will be canceled in this iteration.

The parameters for the merged particle p_k can be a weighted sum of that for p_i and p_j according to the mass. By a weighted sum computation, we can correctly handle the merge operation for particles in different level or with different material.

3.5 Flowing Control System

So far, our method provides an reasonable fluid simulation framework and a thermal transfer scheme for simulating the heat flow, which can trigger the melting event. Although we have the ability to simulate the object melting, the animators may expect to design the flowing pattern for object melting, including the shape of the fluid. To this end, we develop a flowing control system, aiming to provide two types of control mechanism for different purpose. The first one is called flowing path control, which is used for guiding the direction of melting flow. The second one is named as melting shape control, which generate an approximate geometric potential field to constrain the flow shape. We utilize the control particles concept as in [TKPR06] to guide the fluid volume.

Figure 3.10 shows the control system flowchart in our approach.

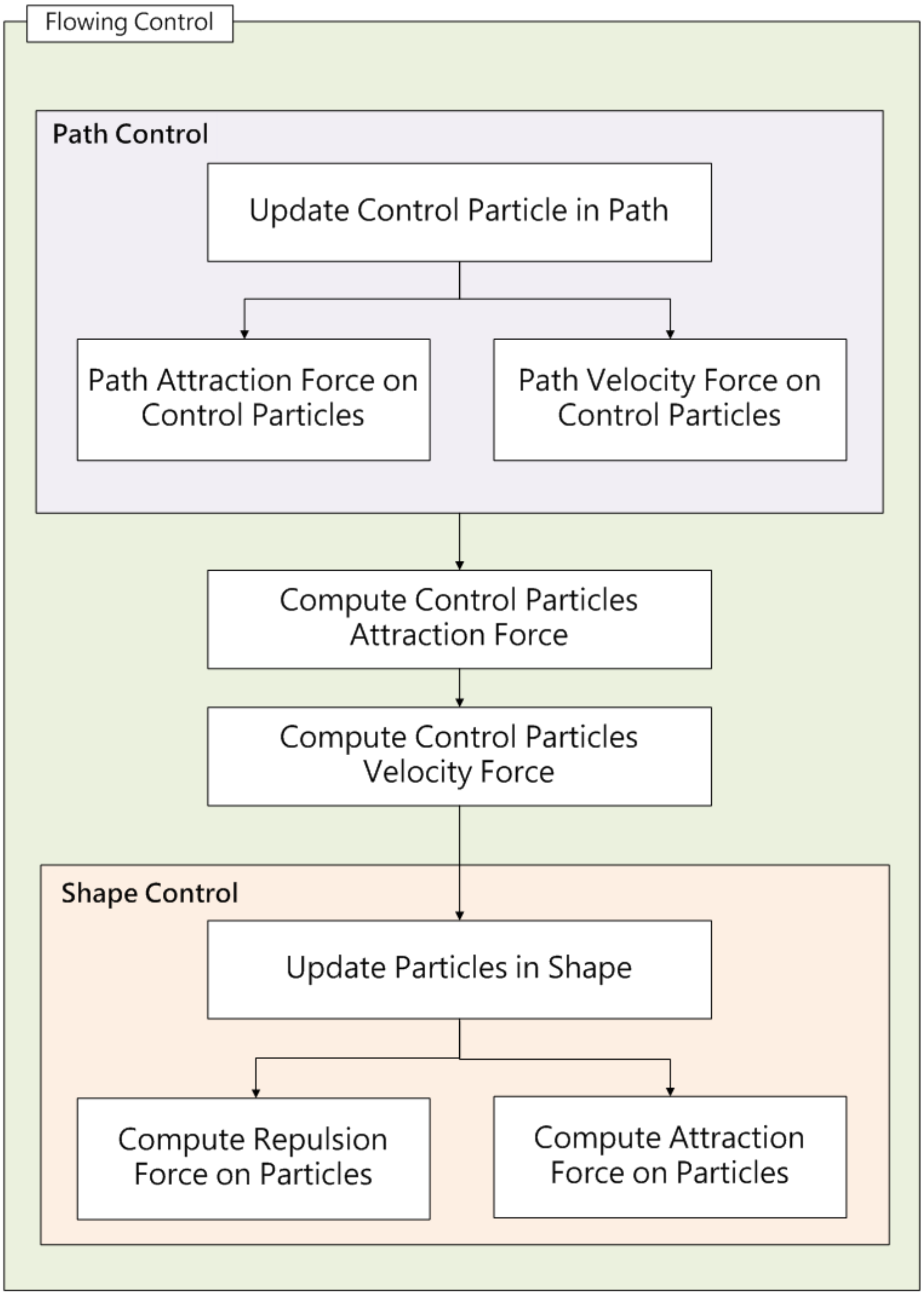
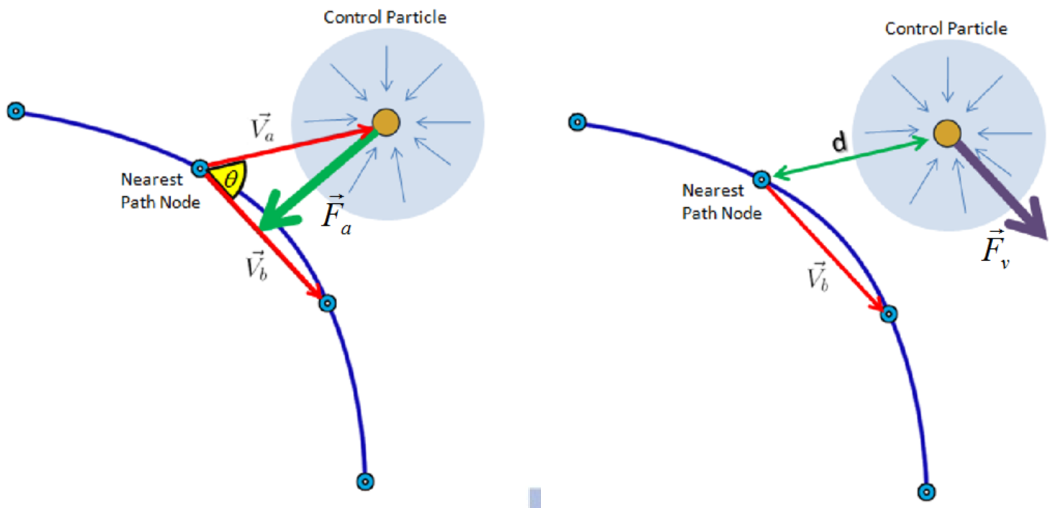

Flowing Control
Path Control
Update Control Particle in Path
Path Attraction Force on Control Particles
Path Velocity Force on Control Particles
Compute Control Particles Attraction Force
Compute Control Particles Velocity Force
Shape Control
Update Particles in Shape
Compute Repulsion Force on Particles
Compute Attraction Force on Particles

Figure 3.10: Flowing control flowchart.

At the initial stage, we sample a set of control particles through the SPH particles. We define an influence range for each control particle to control the nearby particles. Each control particle has two forces to affect the movement of SPH particles. Attraction force drags the nearby particles to the control particles and the velocity force gives the nearby particles the forces based on the control particle's moving direction. The strength of force is inversely proportional to the distance between the control particle and other SPH particles. We make use of these control particles to guide all the SPH particles move as the user designed. Furthermore, we also utilize the attraction force to enhance small droplet formation during the melting process.

3.5.1 Path Control

For path control, user can draw several flowing paths directly on the object surface or other constrainted surface. These path are used for guiding the nearby control particles to flow along the path. Instead of apply forces on all the normal SPH particles, we consider the control particles, which is a subset of SPH particles, are enough to guide the fluid direction. Moreover, by lowering the strength of force with respect to the distance to the center of control particle, more fluid detail can be preserved. Each path is defined by a set of path nodes generated along a user-draw trajectory, the path nodes have the information about the influence range R, and two forces direction: attraction force and velocity force, which can be used to drag the control particles close to the path and give the control particles a direction along the path. In the end, each path may give the forces to control particles that will move along the path trajectory.



(a) Attraction force (b) Velocity force

Figure 3.11: Two path control forces.

Attraction Force Our method compute the attraction force based on the vector V_a and V_b, which is shown in Figure 3.11(a). The control particle is attracted by the path along the direction $-V_a \sin\theta$, the angle θ is the angle between V_a and V_b. Equation 3.27 is the final attraction force for each control particle:

$$F_a = \sum_i (-\vec{V_a} \sin\theta)_i \tag{3.27}$$

where i indicates the index of each path, F_a is the attraction force.

Velocity Force Velocity force provides the necessary force for fluid to flow along the predefined path. Figure 3.11 (b) represents how we generate the velocity force for each control particle. The path direction is decided by the vector from the current node to the next node position, and the magnitude of velocity force is proportional to the distance between the control particle and the path node. Equation 3.28 shows the formulation we use in our method:

$$F_v = \sum_i (\vec{V_b} W(d, R))_i \tag{3.28}$$

where i is the index of path, $\vec{V_b}$ is the direction along the path, $W(d, R)$ is a smoothing function (e.g. gaussian function) with influence range R defined on the path.

The final force for each control particle can be get as a weighted sum of F_a and F_v as shown in Equation 3.29.

$$F_{path} = (W_a * F_a + W_v * F_v) \tag{3.29}$$

In addition, we apply a feedback control scheme to adjust the force strength during simulation. Considering the fluid force acting on the control particles, which is computed by the fluid solver in 3.2, if this fluid force direction is away from the control direction, we need a larger control force to correct the flowing direction. The strength of path control is proportional to the angle θ_p between the force direction provided by the fluid solver and the force direction provided by the paths. Equation 3.30 is the force provided by a path.

$$F_{Path_f} = W_p * F_{path} \tag{3.30}$$

Equation 3.31 shows the weighting coefficient W_p computation as a feedback scheme, where k_p is a constant coefficient and θ_p in Equation 3.32 is the angle between the force direction provided by the fluid solver and the force direction provided by the paths.

$$W_p = k_p(-\theta_p + 1.0) \tag{3.31}$$

$$\theta_p = \frac{F_{fluid} \cdot F_{path}}{||F_{fluid}|| \, ||F_{path}||} \tag{3.32}$$

where F_{fluid} is the force direction provided by the fluid solver and F_{path} is the force direction provided by the paths.

Except for the force control we have mentioned above, our path control also controls the temperature absorbtion rate to achieve our goal. Since the fluid will go toward low to flow, we slightly change the heat adoption coefficient δ at starting point of the path, so the melting may happen faster than other regions, causing the melting flow toward our control path. The heat adoption coefficient is simply modified by a Gaussian function, making the adoption coefficient smoothly varying according to the distance to the center of path point. Figure 3.12 shows the adoption coefficients varies in a color-coded particles diagram.

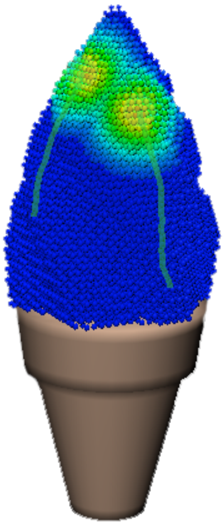


Figure 3.12: Modify the adoption coefficient around the path starting point

3.5.2 Shape Control

It is quite often that artists have the needs to control the fluid to form a target shape. In our object melting simulation, we provide an easy-design shape control interface for artists to do these jobs. Differ from path control scheme, shape control must control all the SPH particles more precisely to obey the constraints we set. Because user will not expect there are small droplet cross the shape or there are some details do not follow this constraint. Therefore, instead of only applying on control particles, we enforce all the SPH particles in the influence range of control shape to accept the control forces.

As the same way in path creation, we allow user to draw shape contours on the scene surface, and store each contours with discrete nodes along the trajectory. Although the shape contour is 2D on the surface, the influence is still on a 3D volume around each node. We think this is adequate for a melting process control since the fluid is usually flowing close to the surface. A shape contour has several shape nodes on the trajectory. And for each shape node, it stores a direction for the contour repulsion force and two influence range as shown in Figure 3.14. The farther range is the attraction range used to attract SPH particles to move close the contour line and the closer range is the repulsion range used to keep the particles stay inside the shape contour region.

Once a shape contour is drawn, we then compute the shape force direction for each node on the shape contour. By finding a nearest surface vertex normal vector and the vector to its next node position, we compute the shape force vector as the cross product of these two vectors, as shown in Figure 3.13.

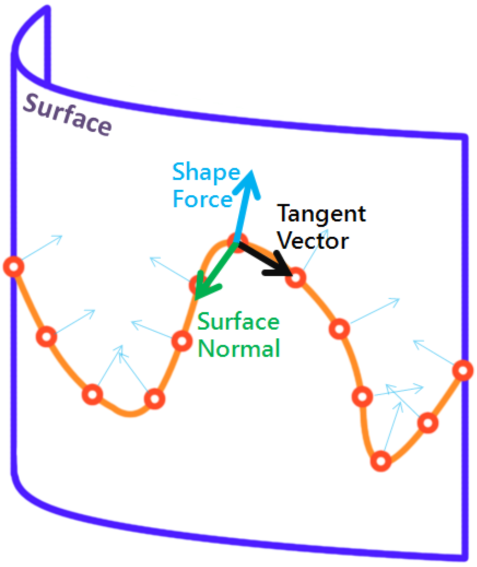


Figure 3.13: Shape control force for each node.

We divide the influence range for each node into two level: attraction and repulsion range. Figure 3.14 reveals the two level range in our approach. Inside the light red region, we employ the shape force describe above to enforce the fluid inside the shape contour. In the light blue region, we use a attraction force to attract the nearby particles to form our target shape, and the direction is simply the vector from particle to shape node.

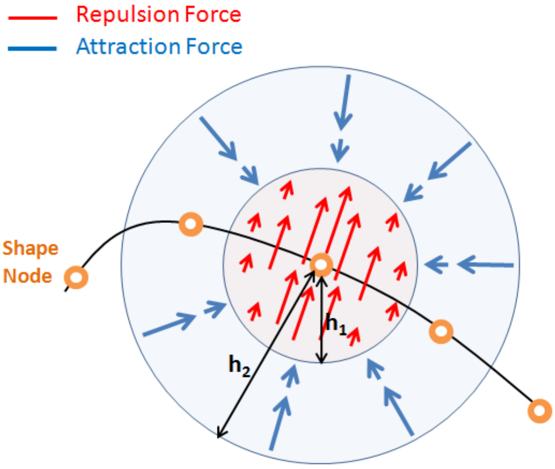


Figure 3.14: Shape node two level influence range

Equation 3.33 shows the formulation of shape attraction force, which is used to attract the nearby SPH particles to form the target shape.

$$F_{Sa} = C_{Sa} V_{Sa} W(d - h_1, h_2 - h_1) \tag{3.33}$$

where F_{Sa} is the attraction force, C_{Sa} is the coefficient of attraction strength, V_{Sa} is the attraction direction from particle position to the shape node position, and W is a smoothing kernel to smooth interpolate the magnitude of the force with the distance d.

The repulsion force computation can be done as Equation 3.34:

$$F_{Sr} = C_{Sr} V_{Sr} W(d, h_1) \tag{3.34}$$

where F_{Sr} is the repulsion force, C_{Sr} is the coefficient of repulsion strength, V_{Sr} is the repulsion direction for the shape node, and W is a smoothing kernel to smooth interpolate the magnitude of the force with the distance d.

Moreover, we also control the temperature of particles within the influence region. We slightly lower down the temperature per iteration until it is below to the melting point, so the particle will stick to the target shape we design.

3.6 Implementation Details

For the particle moving, we integrate the particle velocity, position by using an explicit Euler scheme (Equation 3.35, 3.36) for a given timestep. Equation 3.35 shows the computation of velocity update in an explicit Euler scheme:

$$v_{t+\Delta t} = v_t + \Delta t * (\frac{p_F}{p_m}) \tag{3.35}$$

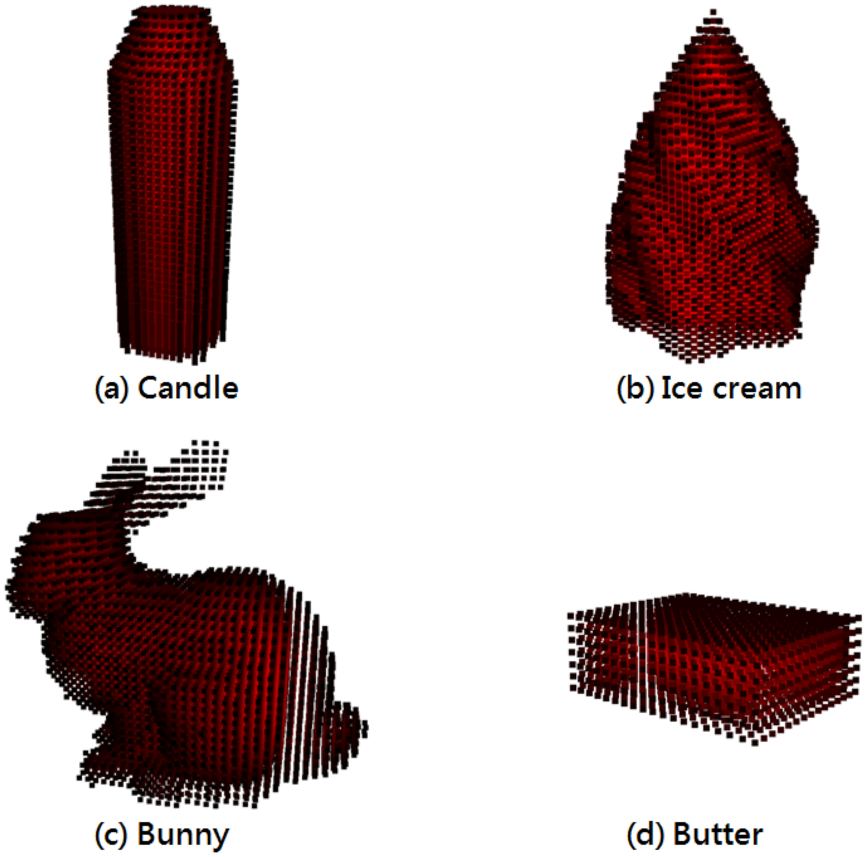
where $v_{t+\Delta t}$ is the velocity in next iteration, Δt is the time interval, p_F is the force acting on particle, and p_m is the mass of particle.

Equation 3.36 shows the computation for updating the particle position.

$$r_{t+\Delta t} = r_t + \Delta t * p_v \tag{3.36}$$

where $r_{t+\Delta t}$ is the particle position in next iteration, p_v is the velocity of the particle.

To speedup the neighborhood searching, we build a k-d tree per iteration to eliminate the farther particles quickly. The obstacle of the scene and the initial fluid volume are all represented as signed distance fields [FPRJ00]. Figure 3.15 shows four different models of the signed distance fields, we only draw the distance larger than zero. The color indicates the distance value to the object surface from black to red.


(a) Candle

(b) Ice cream

(c) Bunny

(d) Butter

Figure 3.15: Signed distance value in four different object

In the real world, some of the materials will change the phases quickly. For instance, the wax of the candle will directly burn out instead of change into the liquid phase. Our system also take into account of this circumstance, we randomly delete the high temperature particles to decrease the fluid generation rate. The possibility of deleting particles is depending on the material properties.

For the surface rendering, we employ the distance based surface tracking method to define the surface induced by particles [APKG07]. Then we extract a triangle mesh by using the marching cube algorithm [LC87]. In some part of our implementation code, we make use of the parallelization of multi-core CPU by applying OpenMP library. This could saves almost one-third of the computational time in our dual core CPU testing platform.

CHAPTER 4

Results

In this chapter, we are going to present the experiment results for different simulation scenes. We list the particle number and the computational time for each iteration as a table below, and we also compare our candle melting result with the [PPLT09]. In the end, we show several melting scenes with different materials, and some of them are produced with the effect of control sketches painting by users. All the experiments are performed on Intel core 2 duo CPU E6750 2.66GHz with 4GB ram, and the video card is Nvidia GTX260.

4.1 Simulation Statistics

Our adaptive scheme will dynamically alter the particle number during the simulation, we summarize the related statistics in Table 4.1.

Scene	Min particle	Avg min time($\frac{sec.}{iter.}$)	Max particle	Avg max time($\frac{sec.}{iter.}$)
Candle	8017	1.577	16427	6.649
Ice Cream	3724	0.743	15399	7.574
Chocolate Bunny	4356	1.486	17685	11.338
Butter	2640	0.484	2640	0.484

Table 4.1: Simulation statistics of different scenes

Though the particle number in candle, ice cream, chocolate bunny are nearly the same, the average time are not close to each other. This is due to the number of particle in thermal simulation are different, the more particle need to simulate the heat energy transfer, the longer computation time it will cost. Table 4.2 lists the percentage of time consumption in the fluid solver and thermal transfer stages. It explains the reason in Table 4.1, which the particle number in thermal simulation stage plays an critical role in our computation cost.

Scene	Fluid Solver	Thermal Simulation
Candle	25%	75%
Ice Cream	16%	84%
Chocolate Bunny	12%	88%
Butter	40%	60%

Table 4.2: The percent of time consuming in different stages

4.2 Simulation Results

In this section, we shows the simulation results in several parts, the results are displayed by a sequence of pictures. In subsection 4.2.2, we shows the heat energy propagation between particles by representing the temperature in color. It is clear that the material properties are changing during the heat up process. Subsection 4.2.3 shows the control effect create by users, including the path control and the shape control. And subsection 4.2.4, we compare our results to previous methods by simulating the same type of material melting. In the end, subsection 4.2.1 exhibit the rest of melting scene we have simulated.

4.2.1 Object Melting Scenes

We show two of the melting scenes we have simulated here: soft ice-cream melting and a chocolate bunny melting. In these two demo, we use two heat sources for giving heat energy. Figure 4.1 shows the melting process of the ice cream respectively. And Figure 4.2 displays the chocolate bunny melting situation.

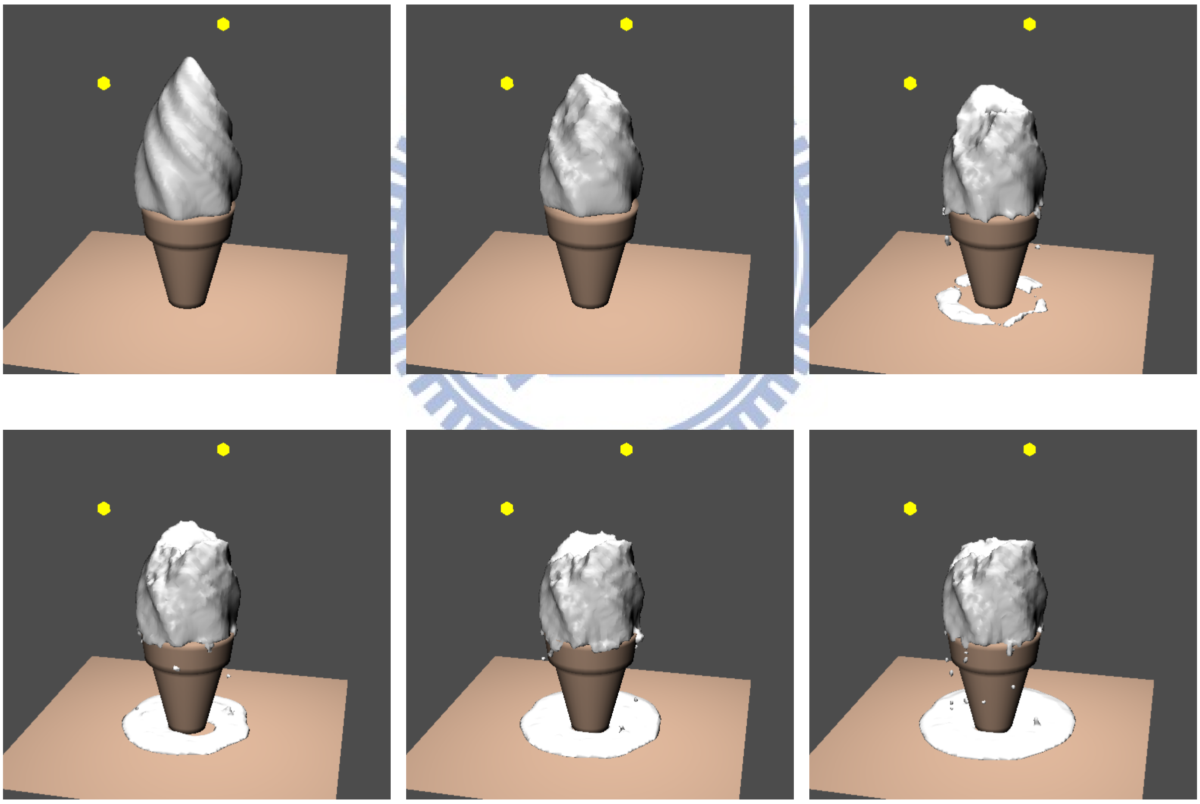


Figure 4.1: Ice cream melting

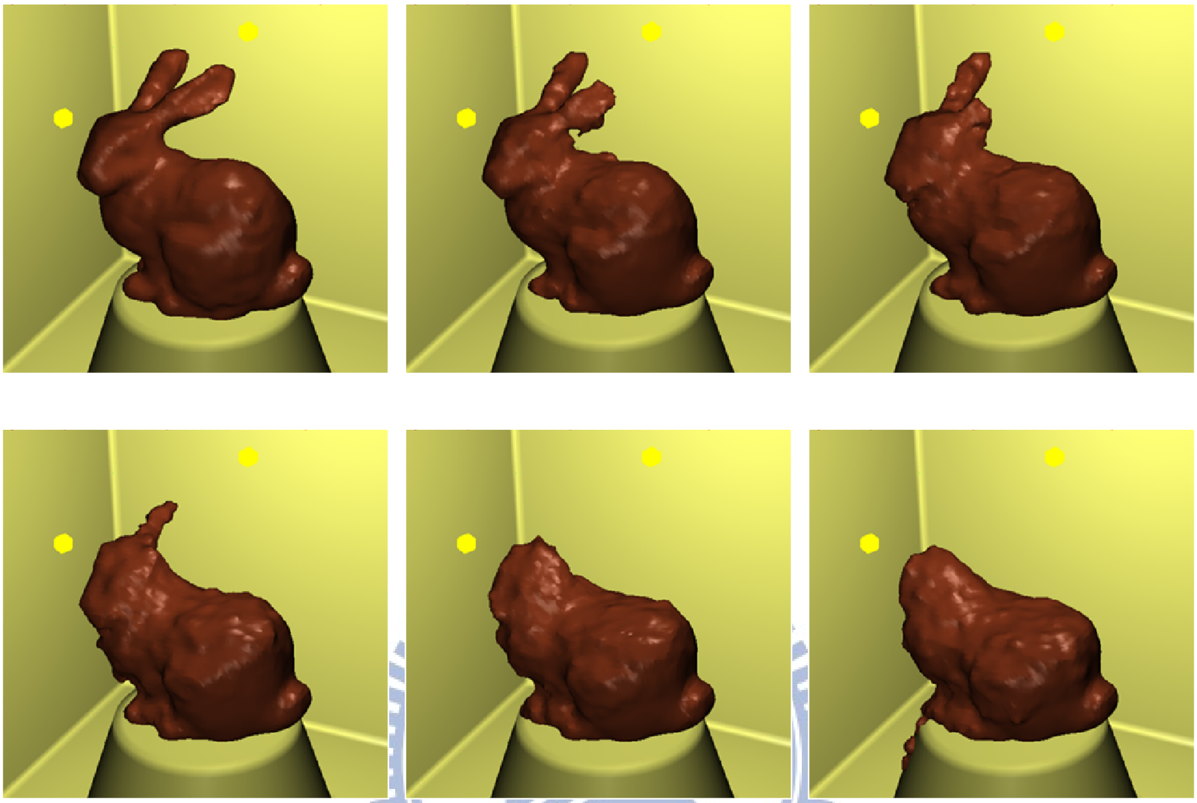


Figure 4.2: Chocolate bunny melting

The results here shows our method is capable to deal with different materials, and produce a reasonable melting behavior compare to the real scene.

4.2.2 Thermal Transfer Simulation Results

The following results shows the thermal simulation temperature in each particle, the color from blue to red represents the temperature from low to high. In the results, we can clearly see the heat energy propagate within the particles from surface particles to inside volume. And the particles close to heat sources may heat up faster than the particles are farther or be occluded.

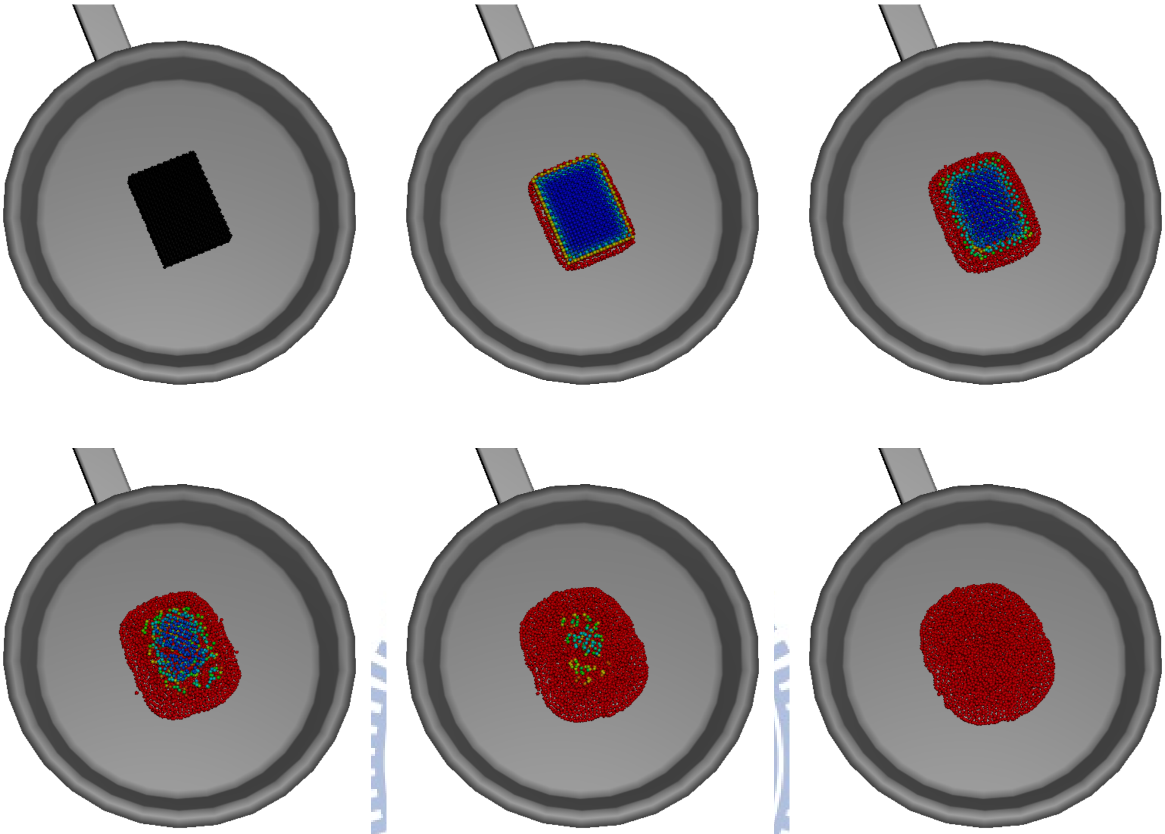


Figure 4.3: Thermal simulation in butter scene

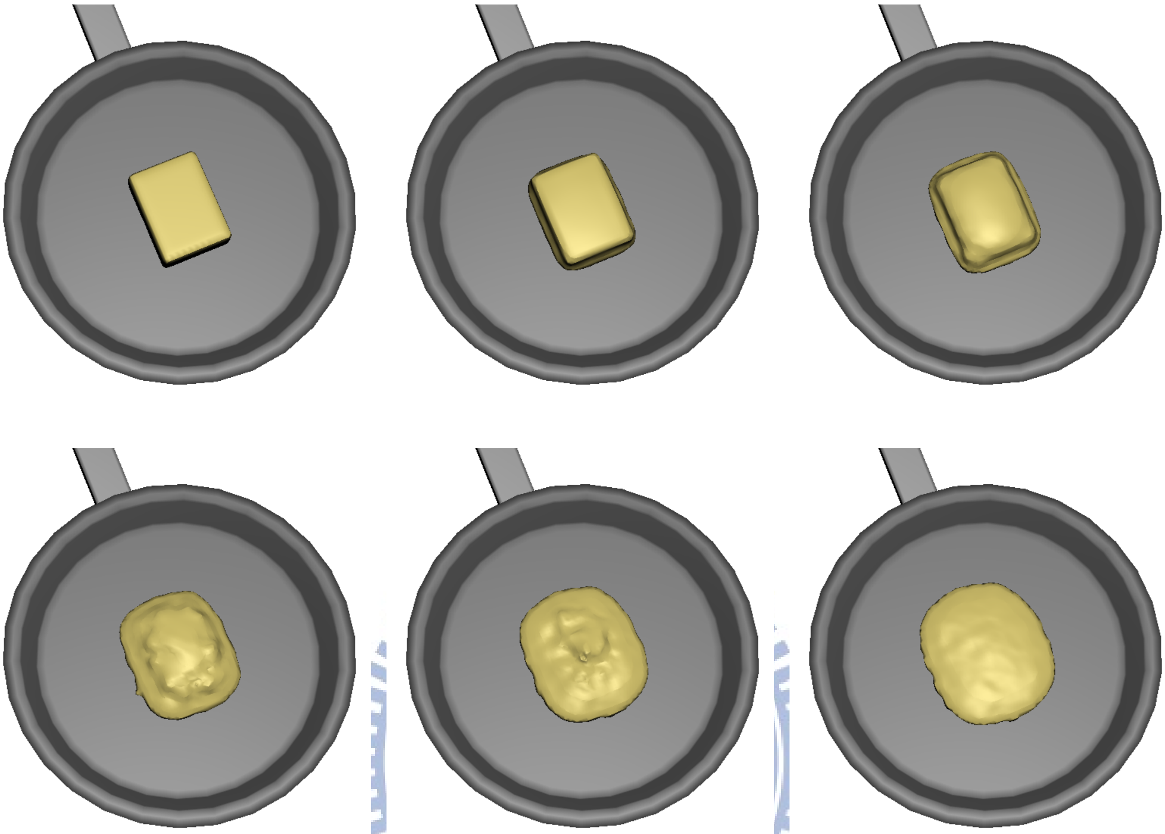


Figure 4.4: Final image in butter scene

4.2.3 Fluid Control Results

Here we show the results of two types control forces in our method. In Figure **??**, we make use of the path control force to guide the melting flow movement. We carefully set the control parameter to avoid the unreal fluid flowing and droplet spray. As in the result, by applying the path control, the melting fluid will intend to flow along the direction which provide by users.

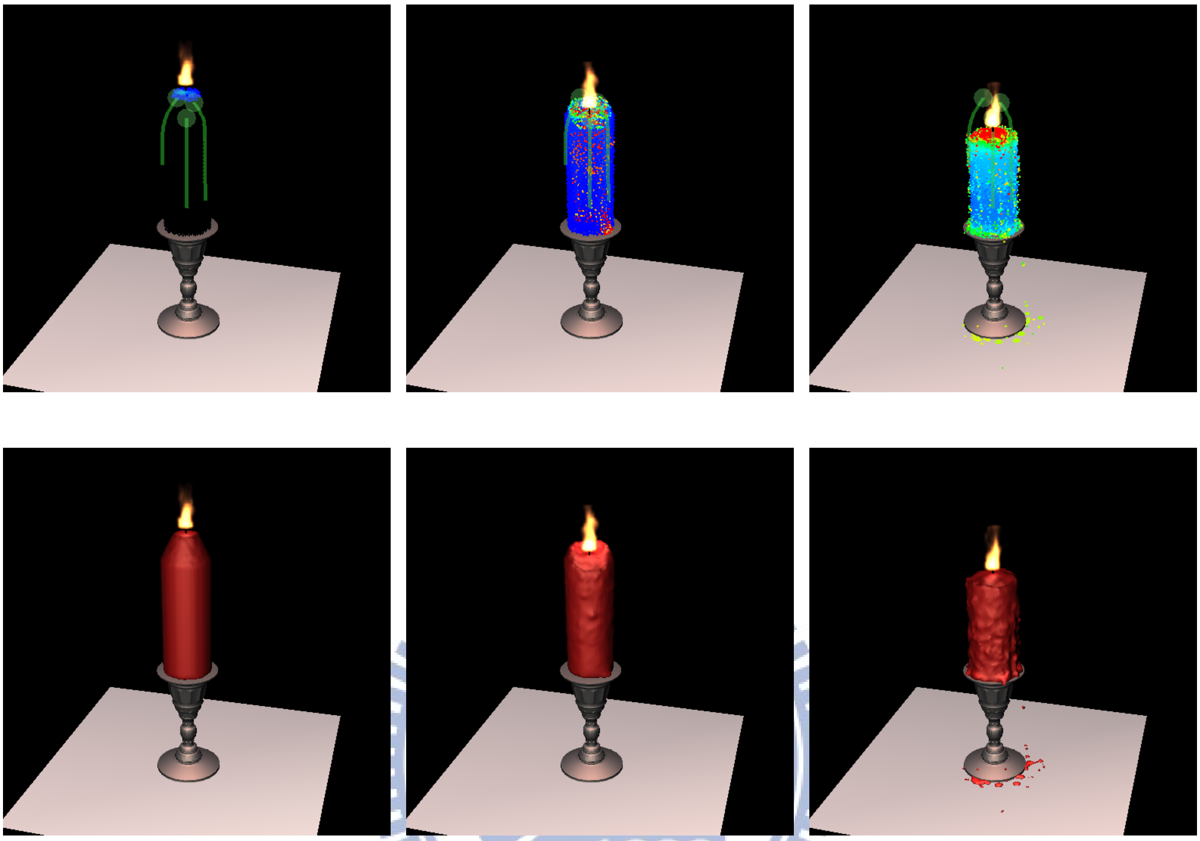


Figure 4.5: Candle melting in path control

The shape force control provides us the ability to design the overall shape of the melting fluid. Figure 4.7 shows a control example in butter scene, where user draws a smiling face as a target shape, and during the melting simulation time, the melting fluid will fill the target shape smoothly.

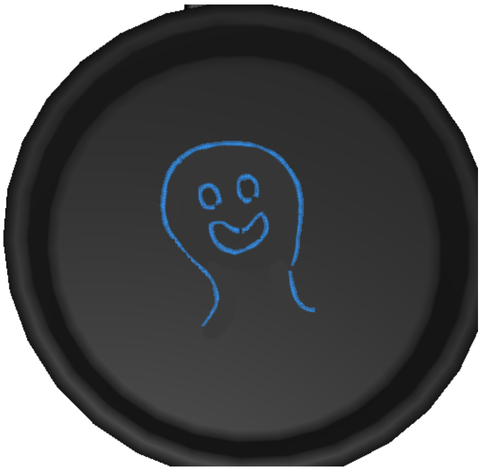


Figure 4.6: The shape control line drawn by user

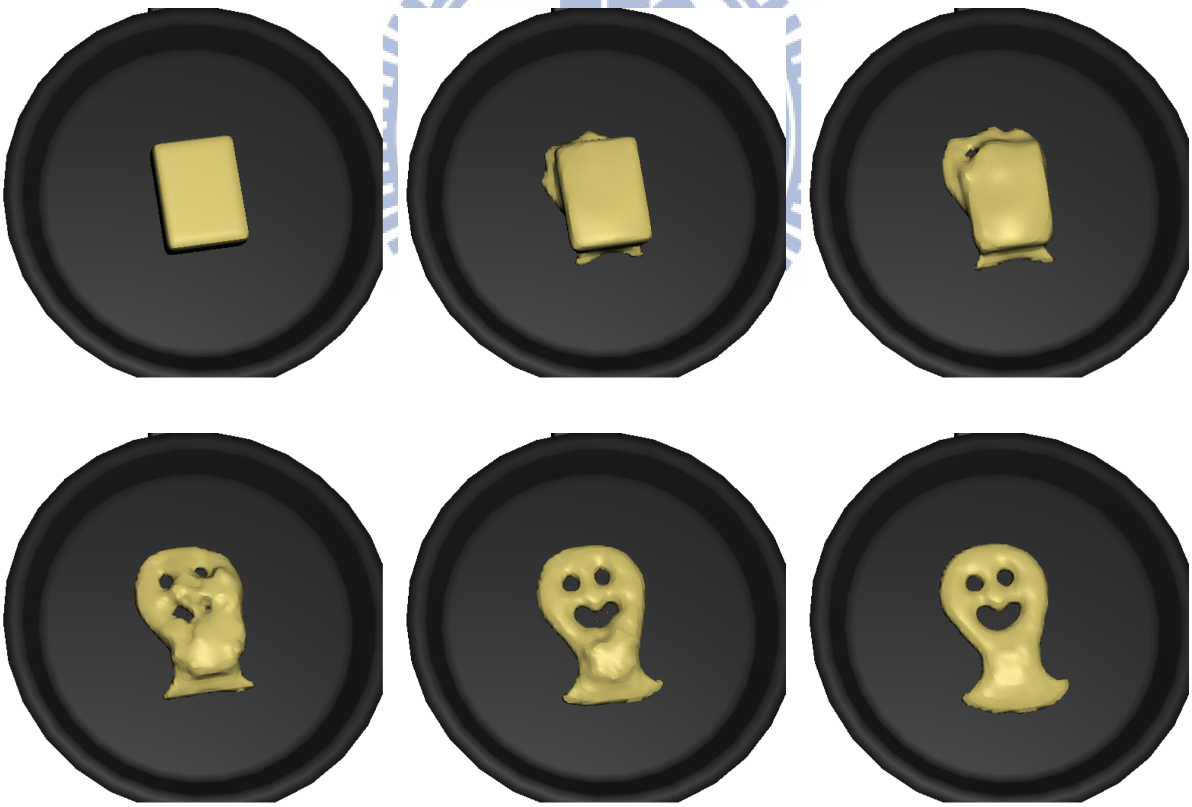


Figure 4.7: Butter melting in shape control

4.2.4 Comparison with Previous Methods

Figure 4.8, 4.9, 4.10 shows a comparison of a candle melting scene between our result, Carlson et al. [CMVHT02], and Pavia et al.[PPLT09]. Through our thermal transfer works, the heat flow inside the particles can be simulated more naturally. Also, the attraction force of control particles slightly encourage the formation of detail droplet, make our result more realistic than the previous works.

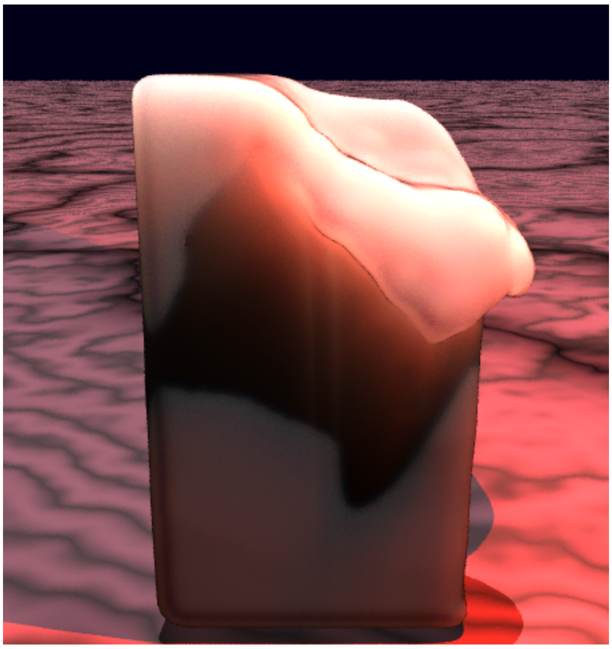


Figure 4.8: Wax melting in [CMVHT02]

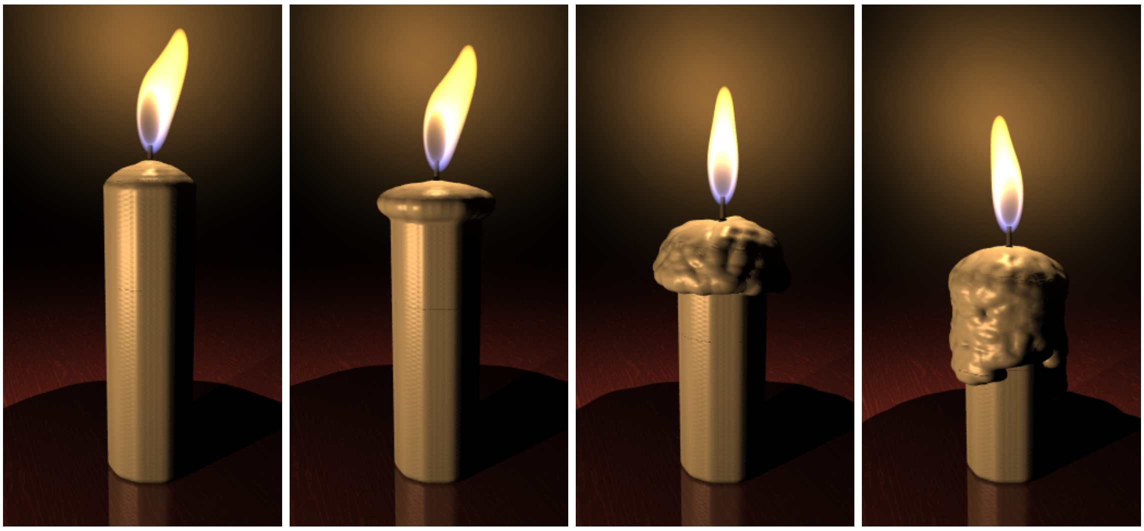


Figure 4.9: Candle melting in [PPLT09]

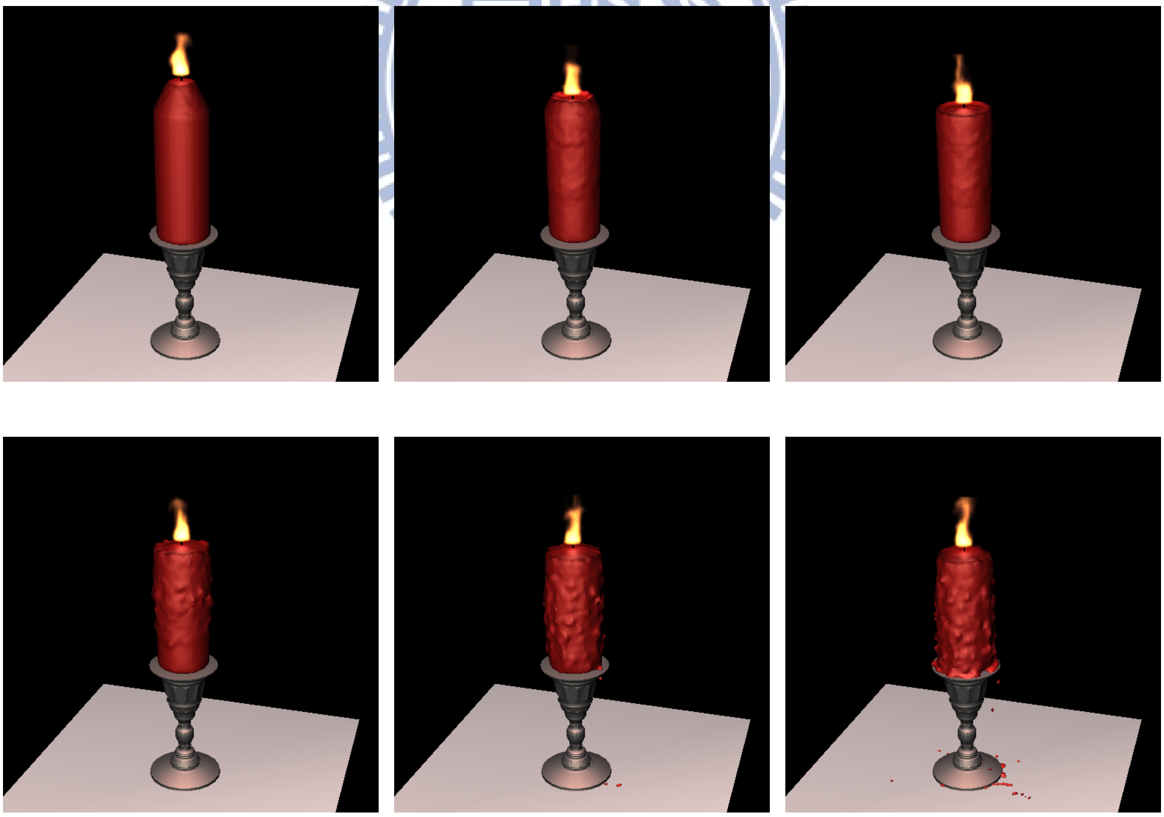


Figure 4.10: Candle melting in our approach

CHAPTER 5

Conclusions

In this chapter, we brief summarize our object melting and flowing control approach algorithm. Furthermore, we analyze the limitations and purpose some future directions to improve our works.

5.1 Summary

In summary, we have purpose a particle-based fluid solver and a physics-inspired thermal transfer system to simulate the heat energy flow in fluid particles. By adding a elastic stress tensor to the traditional Navier-Stokes equation, our method can easily handle the high viscosity fluid behavior, which is often seen in melting scene. Also, our method take advantage of the adaptive particle sampling scheme to accelerate the simulation speed. In thermal transfer system, we have simulated three types heat energy transfer manner: thermal radiation, thermal conduction, and thermal dissipation. Through these transfer sways, the heat energy can be propagate to neighboring particles naturally, make us able to trigger the melting event happening based on the temperature in fluid elements.

Except for the physics-based simulation, we also pay attention to the controllability which is curial for animators. We utilize the control particles to guide the whole fluid volume flowing direction, which may preserve more fluid detail from the fluid solver. And we design two types of control mechanism, the path control can lead the fluid flow along the path, and the shape control has the ability to constraint the melting fluid to form our desire shape automatically. Besides the force control for particle movement, we additional control the temperature variables to achieve our goal during the melting process.

The results demonstrate that our method can melting several different materials naturally. And the animators can easily design the melting scene by the intuitive sketching tools.

5.2 Limitations and Future Work

Our object melting and flowing control framework is able to deal with a wide range of materials, and provide a realistic melting process. But there are still some limitations in our approach. For real objecting melting process, there are plenty of details (e.g. small droplet) flowing on the surface. Current object surface in our framework is reconstructed by marching cube algorithm, it is hard to avoid some small details are smoothing during the polygon creation. Moreover, since our control mechanism is a force-based system, it is difficult to control each particle to the the target position precisely. Finally, we have not try to simulate inhomogeneous materials in this thesis. The inhomogeneous material contains different material types in a single object, which makes the interaction between particles more complicate.

In the future, we would like to enhance the surface detail by applying more recently surface reconstruction method, such as adaptive marching cube or other advance techniques. Besides, currently our computational cost bottleneck is on the thermal simulation part, more particle number in thermal exchanging stage may lower down the simulation speed. To apply a important sampling scheme in thermal simulation maybe a good idea to improve the efficiency. The region of interest may use more samples to capture the accurate thermal transfer result, and the other region may use less particles for a sufficient simulation. Lastly, we are now focusing on extending the path control works to form a 3D target. The path can be done by some mesh information like skeleton, and the nodes can store local volume inside the mesh and the contour around the path node. Hopefully, this control mechanism could help the melting fluid to form any given target shape.

Bibliography

[APKG07] B. Adams, M. Pauly, R. Keiser, and L. J. Guibas. Adaptively sampled particle fluids. In *SIGGRAPH '07: ACM SIGGRAPH 2007 papers*, page 48, New York, NY, USA, 2007. ACM.

[CBL+09] Y. Chang, K. Bao, Y. Liu, J. Zhu, and E. Wu. A particle-based method for viscoelastic fluids animation. In *VRST '09: Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology*, pages 111–117, New York, NY, USA, 2009. ACM.

[CMVHT02] M. Carlson, P. J. Mucha, R. B. Van Horn, III, and G. Turk. Melting and flowing. In *SCA '02: Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 167–174, New York, NY, USA, 2002. ACM.

[DC96] M. Desbrun and M. Cani. Smoothed particles: A new paradigm for animating highly deformable bodies. In Ronan Boulic and Gérard Hégron, editors, *Eurographics Workshop on Computer Animation and Simulation, EGCAS '96, August, 1996*, pages 61–76, Poitiers, France, 1996. Springer-Verlag. Published under the name Marie-Paule Gascuel.

[DG96] M. Desbrun and M. Gascuel. Smoothed particles: A new paradigm for animating highly deformable bodies. In *In Computer Animation and Simulation*

96 (Proceedings of EG Workshop on Animation and Simulation, pages 61–76. Springer-Verlag, 1996.

[DKNY08] Y. Dobashi, K. Kusumoto, T. Nishita, and T. Yamamoto. Feedback control of cumuliform cloud formation based on computational fluid dynamics. *ACM Trans. Graph.*, 27(3):1–8, 2008.

[FF01] N. Foster and R. Fedkiw. Practical animation of liquids. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 23–30, New York, NY, USA, 2001. ACM.

[FL04] R. Fattal and D. Lischinski. Target-driven smoke animation. *ACM Trans. Graph.*, 23(3):441–448, 2004.

[FPRJ00] S. F. Frisken, R. N. Perry, A. P. Rockwood, and T. R. Jones. Adaptively sampled distance fields: a general representation of shape for computer graphics. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 249–254, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[GBO04] T. G. Goktekin, A. W. Bargteil, and J. F. O'Brien. A method for animating viscoelastic fluids. *ACM Trans. Graph.*, 23(3):463–468, 2004.

[HK04] J. M. Hong and C. H. Kim. Controlling fluid animation with geometric potential: Research articles. *Comput. Animat. Virtual Worlds*, 15(3-4):147–157, 2004.

[KAG$^+$05] R. Keiser, B. Adams, D. Gasser, P. Bazzi, P. Dutre, and M. Gross. A unified lagrangian approach to solid-fluid animation. *Proceedings Eurographics/IEEE VGTC Symposium Point-Based Graphics*, 0:125–148, 2005.

[LAD08] T. Lenaerts, B. Adams, and P. Dutré. Porous flow in particle-based fluid simulations. *ACM Trans. Graph.*, 27(3):1–8, 2008.

[LC87] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH Comput. Graph.*, 21(4):163–169, 1987.

[LD09] T. Lenaerts and P. Dutré. Mixing fluids and granular materials. *Comput. Graph. Forum*, 28(2):213–218, 2009.

[MCG03] M. Müller, D. Charypar, and M. Gross. Particle-based fluid simulation for interactive applications. In *SCA '03: Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 154–159, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.

[MGG$^+$10] N. Maréchal, E. Guérin, E. Galin, S. Mérillou, and N. Mérillou. Heat transfer simulation for modeling realistic winter sceneries. *Computer Graphics Forum*, 29(2):449–458, 2010.

[Mon05] J. J. Monaghan. Smoothed particle hydrodynamics. *Reports on Progress in Physics*, 68(8):1703–1759, August 2005.

[MTPS04] A. McNamara, A. Treuille, Z. Popović, and J. Stam. Fluid control using the adjoint method. *ACM Trans. Graph.*, 23(3):449–456, 2004.

[PPLT06] A. Paiva, F. Petronetto, T. Lewiner, and G. Tavares. Particle-based non-newtonian fluid animation for melting objects. In *Sibgrapi 2006 (XIX Brazilian Symposium on Computer Graphics and Image Processing)*, pages 78–85, Manaus, AM, october 2006. IEEE.

[PPLT09] A. Paiva, F. Petronetto, T. Lewiner, and G. Tavares. Particle-based viscoplastic fluid/solid simulation. *Comput. Aided Des.*, 41(4):306–314, 2009.

[REN$^+$04] N. Rasmussen, D. Enright, D. Nguyen, S. Marino, N. Sumner, W. Geiger, S. Hoon, and R. Fedkiw. Directable photorealistic liquids. In *SCA '04: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer* *animation*, pages 193–202, Aire-la-Ville, Switzerland, Switzerland, 2004. Eurographics Association.

[SP09] B. Solenthaler and R. Pajarola. Predictive-corrective incompressible sph. In *SIGGRAPH '09: ACM SIGGRAPH 2009 papers*, pages 1–6, New York, NY, USA, 2009. ACM.

[SY05] L. Shi and Y. Yu. Taming liquids for rapidly changing targets. In *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 229–236, New York, NY, USA, 2005. ACM.

[TKPR06] N. Thürey, R. Keiser, M. Pauly, and U. Rüde. Detail-preserving fluid control. In *SCA '06: Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 7–12, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association.

[TMPS03] A. Treuille, A. McNamara, Z. Popović, and J. Stam. Keyframe control of smoke simulations. *ACM Trans. Graph.*, 22(3):716–723, 2003.

[WLK03] X. Wei, W. Li, and A. Kaufman. Melting and flowing of viscous volumes. In *CASA '03: Proceedings of the 16th International Conference on Computer Animation and Social Agents (CASA 2003)*, page 54, Washington, DC, USA, 2003. IEEE Computer Society.